Convergence Analysis of a Schwarz Type Domain Decomposition Method for the Solution of the Euler Equations

V. Dolean, Stephane Lanteri, Frédéric Nataf

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Convergence analysis of a Schwarz type domain decomposition method for the solution of the Euler equations

V. Dolean, S. Lanteri and F. Nataf

N° 3916

Avril 2000

THÈME 4

ISSN 0249-6399 ISRN INRIA/RR--3916--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Convergence analysis of a Schwarz type domain decomposition method for the solution of the Euler equations

V. Dolean*, S. Lanteri* and F. Nataf †

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet Sinus

Rapport de recherche n° 3916 — Avril 2000 — 45 pages

Abstract: we report on a preliminary convergence analysis of a domain decomposition method for solving the Euler equations for compressible flows. This method was previously described in Dolean and Lanteri[3]. It relies on the formulation of an additive Schwarz type algorithm on a non-overlapping decomposition of the computational domain. According to the hyperbolic nature of the Euler equations, the transmission conditions that are set at subdomain interfaces, express the conservation of the normal flux. In [3], this method is assessed experimentally in the context of a flow solver which is based on a mixed finite volume/finite element formulation on unstructured triangular meshes. Here, we study the convergence of the proposed method in the two- and three-dimensional cases, and for a two-subdomain decomposition, by considering the linearized equations and applying a Fourier analysis. In doing so, we observe that in spite of the fact that we use simple transmission conditions, the method converges and demonstates, for particular flow conditions, an optimal convergence rate. Various numerical experiments allow to exhibit at least qualitatively, the convergence behavior obtained analytically.

Key-words: Domain decomposition method - Euler Equations - Finite volumes - Triangular meshes - Multigrid algorithm - Parallel computing

* INRIA, 2004 Route des Lucioles, BP. 93, 06902 Sophia Antipolis Cédex-France
† CMAP, Ecole Polytechnique, 91128 Palaiseau Cedex (FRANCE)

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Analyse de convergence d'une méthode de décomposition de domaine de type Schwarz pour la résolution des équations d'Euler

Résumé : dans ce rapport nous présentons une analyse preliminaire de convergence d'une méthode par sous-domaine appliquée à la résolution numérique du système des équations d'Euler. Cette méthode a auparavant été décrite dans Dolean and Lanteri[3]. Elle repose sur la formulation d'un algorithme de type Schwarz additif sur une décomposition sans recouvrement du domaine de calcul. En accord avec la nature hyperbolique des équations d'Euler, les conditions de transmission posées aux interfaces entre sous-domaines expriment la continutité du flux normal. Dans [3], cette méthode est évaluée expérimentalement dans le contexte d'un solveur reposant sur une formulation mixte volumes finis/éléments finis en maillages triangulaires non-structurés. Dans ce rapport, nous étudions la convergence de cette méthode dans les cas à deux et trois dimensions d'espace, pour une décomposition en deux sous-domaines, en appliquant une analyse de Fourier aux équations linéarisées. En dépit du fait que les conditions d'interface sont simples, nous observons que la méthode converge et nous démontrons, dans des situations d'écoulement particulières, un taux de convergence optimal. Différentes simulations numériques permettent d'exhiber un comportement qualitativement en accord avec les résultats de l'analyse.

Mots-clés : Méthode de décomposition de domaine - Equations d'Euler - Volumes finis - Maillages triangulaires - Algorithme multigrille - Calcul parallèle

1 Introduction

In this paper we are concerned with the formulation and the analysis of a domain decomposition method for the solution of hyperbolic systems of conservation laws and more particularly, for the solution of the Euler equations that model inviscid compressible flows. In the two-dimensional case the conservative form of these equations writes as :

$$\partial_t W + \partial_x F_1(W) + \partial_y F_2(W) = 0 \tag{1}$$

where the unknown vector of *conservative variables* is given by $W = (\rho\ ,\ \rho u\ ,\ \rho v\ ,\ E)^T$; the hyperbolic fluxes can be written as :

$$F_1(W) = \begin{bmatrix} \rho u \\ -\frac{1}{2}(\gamma-3)\rho u^2 - \frac{1}{2}(\gamma-1)\rho v^2 + (\gamma-1)\rho E \\ \rho u v \\ \rho u(\gamma E - \frac{1}{2}(\gamma-1)(u^2+v^2)) \end{bmatrix}$$

$$F_2(W) = \begin{bmatrix} \rho v \\ \rho u v \\ -\frac{1}{2}(\gamma-1)\rho u^2 - \frac{1}{2}(\gamma-3)\rho v^2 + (\gamma-1)\rho E \\ \rho v(\gamma E - \frac{1}{2}(\gamma-1)(u^2+v^2)) \end{bmatrix}$$

Under the hypothesis that the solution is regular one can also write a non-conservative (or quasi-linear) equivalent form of Eq. (1) :

$$\partial_t W + A_1(W)\partial_x W + A_2(W)\partial_y W = 0 \tag{2}$$

the Jacobian matrices of the flux vectors $A_1(W) = \dfrac{\partial F_1(W)}{\partial W}$ and $A_2(W) = \dfrac{\partial F_2(W)}{\partial W}$ being given by :

$$A_1(W) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2}(\gamma-3)u^2 + \frac{1}{2}(\gamma-1)v^2 & -(\gamma-3)u & -(\gamma-1)v & \gamma-1 \\ -uv & v & u & 0 \\ u\frac{(\gamma-1)(\gamma-2)q^2 - 2c^2}{2(\gamma-1)} & (\frac{3}{2}-\gamma)u^2 + \frac{1}{2}v^2 + \frac{c^2}{\gamma-1} & -(\gamma-1)uv & \gamma u \end{pmatrix}$$

$$A_2(W) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ \frac{1}{2}(\gamma-1)u^2 + \frac{1}{2}(\gamma-3)v^2 & -(\gamma-1)u & -(\gamma-3)v & \gamma-1 \\ v\frac{(\gamma-1)(\gamma-2)q^2 - 2c^2}{2(\gamma-1)} & -(\gamma-1)uv & (\frac{3}{2}-\gamma)v^2 + \frac{1}{2}u^2 + \frac{c^2}{\gamma-1} & \gamma v \end{pmatrix}$$

and where :

$$\begin{cases} q^2 &= u^2 + v^2 \\ c &= \sqrt{\frac{\gamma p}{\rho}} \\ p &= (\gamma-1)\left(E - \frac{\rho}{2}q^2\right) \end{cases}$$

respectively denote the speed of sound and the pressure.

Suppose that we first proceed to an integration in time of (1) using a backward Euler implicit scheme involving a linearization of the flux functions. This operation results in the linearized system :

$$\mathcal{L}U := \frac{1}{\Delta t}U + A_1\partial_x U + A_2\partial_y U = f \tag{3}$$

where $U \equiv W^{n+1} = W(x,(n+1)\delta t)$ and A_1 (respectively A_2) is a shorthand for $A_1(W^n)$ (respectively $A_2(W^n)$).

In the following we are interested in solving problem (3) associated to a suitable set of boundary conditions, by an additive Schwarz type algorithm where the transmission conditions at the interfaces of a non-overlapping decomposition of the computational domain are Dirichlet conditions for the characteristic variables corresponding to the incoming waves, following a strategy already studied by Quarteroni and Stolcis[17]. The novelty in the formulation that is proposed here, is that in the discrete case the interface conditions are expressed in terms of upwind conservative normal fluxes

computed using the approximate Riemann solver of Roe[19]. This choice is before all motivated by the starting point of our study which is given by a flow solver based on a combined finite element/finite volume formulation on unstructured triangular meshes for the spatial discretization. Time integration of the resulting semi-discrete equations is obtained using a linearized backward Euler implicit scheme. As a result, each pseudo time step requires the solution of a sparse linear system for the flow variables, which is the discrete counterpart of (3). Details about the implementation in this numerical framework of the domain decomposition algorithm considered here, are given in Dolean and Lanteri[3].

In section 2 we formulate the additive Schwarz algorithm for a general linear hyperbolic system of equations. In section 3 we apply this algorithm to the solution of the Euler equations and we consider system (3) with frozen coefficients so that we can apply a Fourier analysis in order to derive the convergence rate of the Schwarz algorithm. First, we apply a well-known transformation to reduce the system to a symmetric form. Then, we adopt a decomposition of the (rectangular) computational domain in vertical strips and we apply a Fourier transform in the tangential direction (y direction) to obtain a system of ODEs, for each subdomain, of the form :

$$\begin{cases} \dfrac{d}{dx}\hat{E}_j^{p+1} = -M(k)\hat{E}_j^{p+1} \\ \\ + \text{ Boundary conditions} \\ + \text{ Interface conditions} \end{cases} \tag{4}$$

where $\hat{E}_j$ is the error vector in the Fourier space associated to subdomain j and p denotes the Schwarz iteration. We note that such an approach has already been used for a convection-diffusion equation by Nataf *et al.*[16] and Japhet [10], and for several hyperbolic systems by Kroner[12], Gonzalez[9] (Euler equations) and Clerc[2] (Cauchy-Riemann equations).

In each subdomain, the solution of the resulting system of ODEs can be expressed as a linear combination of the eigenvectors of $M(k)$, the incoming waves being related to the eigenvalues with negative real part (λ_j such that $\Re(\lambda_j) < 0$) and the outgoing ones with those with positive real part (λ_j such that $\Re(\lambda_j) > 0$). The convergence rate of the Schwarz algorithm will be obtained by studying an interface iteration on the cœfficients of these linear combinations. The analysis will be performed in the two-subdomain case, for the two- and three-dimensional Euler equations.

2 Non-overlapping domain decomposition for the Euler equations

In this section we briefly review the main definitions and properties of hyperbolic systems of conservation laws that are of interest to our study. Then we introduce a non-overlapping additive Schwarz type algorithm which is based on transmission conditions at sudomain interfaces that take into account the hyperbolic nature of the problem. Finally, we conclude this section by showing that the Schwarz algorithm can be recasted as an iteration on interface unknowns.

2.1 Hyperbolic systems and boundary conditions

Let us consider a general system of conservation laws of the form :

$$\partial_t W + \sum_{i=1}^{d} \partial_{x_i} F_i(W) = 0, \quad W \in \mathbb{R}^p \tag{5}$$

where d denotes the space dimension and p the dimension of the system. The flux functions F_i are assumed differentiable with respect to the state vector $W = W(x,t)$. In the general case, the flux functions are non-linear functions of W. In addition, if W is assumed regular, system (5) can be written in quasi-linear form :

$$\partial_t W + \sum_{i=1}^{d} \frac{\partial F_i}{\partial W}(W) \partial_{x_i} W = 0 \tag{6}$$

or :

$$\partial_t W + \sum_{i=1}^{d} A_i(W) \partial_{x_i} W = 0 \tag{7}$$

The $A_i(W) = \dfrac{\partial F_i}{\partial W}(W)$ are the Jacobian matrices of the flux functions $F_i(W)$, with respect to W. Recall that system (5) is said to be hyperbolic if, for any unitary real vector $\boldsymbol{n} \in \mathbb{R}^d$, the matrix $\displaystyle\sum_{i=1}^{d} A_i(W) n_i$ is diagonalizable with real eingenvalues. We are particularly interested in the situation where system (5) is integrated in time using a backward Euler implicit scheme involving a linearization of the flux functions. In that case we have :

$$\frac{\delta W}{\delta t} + \sum_{i=1}^{d} \partial_{x_i} \left[\frac{\partial F_i}{\partial W}(W^n) \delta W \right] = -\mathrm{div}(F(W^n)) \tag{8}$$

where $\delta W = W(x, t^{n+1}) - W(x, t^n) = W^{n+1} - W^n$. When δW is assumed regular, we can write the non-conservative form of system (8) :

$$\left[\frac{1}{\delta t} \mathbb{I} + \sum_{i=1}^{d} \partial_{x_i} \left[\frac{\partial F_i}{\partial W}(W^n) \right] \right] \delta W + \sum_{i=1}^{d} \left[\frac{\partial F_i}{\partial W}(W^n) \right] \partial_{x_i} \delta W = -\mathrm{div}(F(W^n)) \tag{9}$$

System (9) can be symmetrized through the multiplication of an operator Σ (see for example Barth[1]) which, for hyperbolic systems admitting an entropy function, is given by the Hessian matrix of this entropy. This operation results in the following first order system :

$$A_0 \delta W + \sum_{i=1}^{d} A_i \partial_{x_i} \delta W = f \tag{10}$$

with :

$$\begin{cases} A_0 &= \Sigma \left[\dfrac{1}{\delta t} \mathbb{I} + \displaystyle\sum_{i=1}^{d} \partial_{x_i} \left[\dfrac{\partial F_i}{\partial W}(W^n) \right] \right] \\ A_i &= \Sigma \left[\dfrac{\partial F_i}{\partial W}(W^n) \right] \\ f &= -\Sigma \mathrm{div}(F(W^n)) \end{cases} \tag{11}$$

Now, let $\boldsymbol{n} = (n_1, ..., n_p)$ denote the outward normal vector to $\partial\Omega$; we define $A_{\boldsymbol{n}} W$ as the normal trace of W on $\partial\Omega$, with $A_{\boldsymbol{n}} = \displaystyle\sum_{i=1}^{d} A_i n_i$. When dealing with boundary conditions, it is well known that one cannot impose all the components of W on the boundary $\partial\Omega$. Instead, the direction of propagation of the information has to be taken into account in order to obtain a well posed initial and boundary value problem (IBVP) for system (10). More precisely, the number and type of boundary conditions that must be imposed on $\partial\Omega$ are deduced from the expression of system (10) in terms of characteristic variables and is related to information entering the domain Ω. A more rigourous discussion of boundary conditions treatment for hyperbolic systems from gas dynamics, in terms of characteristic variables, is for example given in [8] (see also Quarteroni and Valli[18] for a discussion in the context of domain decomposition

algorithms). In order to do so, we decompose the operator $A_{\boldsymbol{n}}$ in positive and negative parts i.e. $A_{\boldsymbol{n}} = A_{\boldsymbol{n}}^+ + A_{\boldsymbol{n}}^-$. Using the diagonalization of $A_{\boldsymbol{n}}$ which writes as $A_{\boldsymbol{n}} = T\Lambda_{\boldsymbol{n}}T^{-1}$ we have :

$$\left\{ \begin{array}{c} A_{\boldsymbol{n}}^{\pm} = T\Lambda_{\boldsymbol{n}}^{\pm}T^{-1} \\ \Lambda_{\boldsymbol{n}}^{\pm} = \mathrm{diag}(\lambda_i^{\pm})_{1\leq i\leq p} \ \text{ with } \ \lambda_i^{\pm} = \dfrac{1}{2}(\lambda_i \pm |\lambda_i|) \\ \text{and with } \ A_{\boldsymbol{n}}^+ W = -A_{-\boldsymbol{n}}^- W \end{array} \right.$$

In order to obtain a well posed IBVP, we have to impose boundary conditions of the form :

$$A_{\boldsymbol{n}}^- W = A_{\boldsymbol{n}}^- g \tag{12}$$

where $A_{\boldsymbol{n}}^-$ is used to select the information entering the domain Ω. Then, a well known result is that the problem :

$$\left\{ \begin{array}{rcl} \mathcal{L}W = A_0 W + \displaystyle\sum_{i=1}^{d} A_i \partial_{x_i} W & = & f, \quad \text{in } \Omega \\ A_{\boldsymbol{n}}^- W & = & A_{\boldsymbol{n}}^- g, \quad \text{on } \partial\Omega \end{array} \right. \tag{13}$$

with $f \in L^2(\Omega)^p$ and $g \in L_A^2(\partial\Omega)$, has a unique solution $W \in \tilde{H}$ (see for example [2]) with :

$$\tilde{H} = \{W \in L^2(\Omega)^p \text{ such that } \sum_{i=1}^{d} A_i \partial_{x_i} W \in L^2(\Omega)^p \text{ and } W|_{\partial\Omega} \in L_A^{1/2}(\partial\Omega)\}$$

$$\text{with } L_A^{1/2}(\partial\Omega) = \{W \text{ such that } \int_{\partial\Omega} |A_{\boldsymbol{n}}| W \cdot W d\sigma < \infty\}$$

This result is used in the next section to formulate a non-overlapping domain decomposition algorithm for the solution of (13).

2.2 Domain decomposition and interface conditions

The domain decomposition approach for solving (13) consists in defining well posed subproblems so that a local solution on a given subdomain Ω_i is the restriction of the global solution on Ω to Ω_i. The sub-problems will inherit the physical boundary conditions of the global problem for the part of $\partial\Omega_i$ which intersects $\partial\Omega$; in addition, appropriate interface conditions have to be added to the definition of the subproblems for the part of $\partial\Omega_i$ which is common to neighboring subdomains. We shall introduce

a non-overlapping domain decomposition algorithm for the following boundary value problem :

$$\begin{cases} \mathcal{L}W \equiv A_0 W + \sum_{k=1}^{d} A_k \partial_{x_k} W = f \ \text{ in } \ \Omega \\ + \text{ Boundary conditions on } \quad \partial\Omega \end{cases} \tag{14}$$

Let $\Omega = \bigcup_{i=1}^{N} \Omega_i$ be a stripwise (for simplicity of presentation) decomposition of Ω and W_i the solution of the local problem :

$$\begin{cases} \mathcal{L}W_i = f|_{\Omega_i} = f_i \\ + \text{ Boundary conditions on } \quad \partial\Omega \cap \partial\Omega_i \\ + \text{ Interface conditions on } \quad \partial\Omega_i \cap \partial\Omega_j \end{cases} \tag{15}$$

Let $\boldsymbol{n_i}$ be the outward normal to $\partial\Omega_i$. The local solution W_i is prolongated by zero on Ω/Ω_i; then a necessary and sufficient condition to insure that $\sum_{i=1}^{N} W_i$ is the solution of the global problem (14) is that the interface conditions on $\Gamma = \partial\Omega_i \cap \partial\Omega_j$ take the form (see [3] for more details) :

$$\begin{cases} A^-_{\boldsymbol{n_i}} W_i + A^+_{\boldsymbol{n_j}} W_j = 0 \\ \text{and} \\ A^+_{\boldsymbol{n_i}} W_i + A^-_{\boldsymbol{n_j}} W_j = 0 \end{cases} \tag{16}$$

2.3 A non-overlapping additive Schwarz algorithm

For simplicity of presentation, we assume that the domain Ω is the $2D$ plane $\Omega = \mathbb{R}^2$ and we consider the case of a non-overlapping decomposition in vertical strips where the subdomains are defined by $\Omega_i =]\gamma_{i-1}, \gamma_i[\times\mathbb{R}$, $2 \le i \le N-1$, $\Omega_1 =]-\infty, \gamma_1[\times\mathbb{R}$, $\Omega_N =]\gamma_{N-1}, \infty[\times\mathbb{R}$. So the outward normal vectors at the interfaces for the subdomain Ω_i are $\boldsymbol{n}_{i,l} = (-1, 0)$ and $\boldsymbol{n}_{i,r} = (1, 0)$. Consequently, $A^-_{\boldsymbol{n}_{i,l}} = -A^+_{\boldsymbol{n}_{i-1,r}} = -A^+$ and $A^-_{\boldsymbol{n}_{i-1,r}} = A^-$. We define a Schwarz type algorithm where the interface transmission conditions are of the form (16); however, according to (12) and (13) we define these interface conditions by selecting the information entering each subdomain. Let $W_i^{(0)}$ denote the initial appoximation of the solution in subdomain Ω_i, then the approximation at the $(p+1)$-th iteration (where p defines the iteration of the Schwarz algorithm) is the solution of the problem :

$$\left\{ \begin{array}{lcll} \mathcal{L}W_i^{(p+1)} & = & f & \text{in } \Omega_i \\ A^+W_i^{(p+1)} & = & A^+W_{i-1}^{(p)} & \text{on } \Gamma_{i,l} \\ A^-W_i^{(p+1)} & = & A^-W_{i+1}^{(p)} & \text{on } \Gamma_{i,r} \end{array} \right. \tag{17}$$

with the convention that $\Gamma_{i,l}$ (respectively $\Gamma_{i,r}$) is the straight line $x = \gamma_{i-1}$ (respectively $x = \gamma_i$). Moreover, we have that $W_i^{(0)} = W_i^n$ and $W_i^{n+1} = W_i^{(P)}$, P being the number of iterations of the above algorithm (n denotes the time step). Clearly, (17) defines an additive Schwarz type algorithm even though its formulation is somewhat unconventional as it is based on a non-overlapping partitioning of the domain Ω. Such algorithms have been extensively studied by Nataf[14] and Nataf *et al.*[16] for convection-diffusion problems. In particular, these authors have considered the use of high-order optimal interface conditions, inspired from the concept of absorbing boundary conditions for unbounded domains[4], for improving the convergence of the Schwarz algorithm. For the hyperbolic type systems Clerc[2] has proved that such algorithms are convergent.

3 Convergence analysis of the Schwarz algorithm

In this section we study the convergence of the proposed additive Schwarz algorithm (17) when applied to the solution of the Euler equations that model inviscid compressible flows.

3.1 The two-dimensional case

3.1.1 Symmetrization

The starting-point is given by the linearized system :

$$\mathcal{L}W := \frac{1}{\Delta t}W + A_1\partial_x W + A_2\partial_y W = f \tag{18}$$

We can transform this system using a matrix T (one can always suppose that such a matrix exists[12]) such that :

$$\left\{ \begin{array}{lcll} B_1 & = & T^{-1}A_1T & = \text{diag}(\sigma_1\ ,\ \sigma_2\ ,\ \sigma_3\ ,\ \sigma_4) \\ B_2 & = & T^{-1}A_2T & \text{is a symmetric matrix} \end{array} \right.$$

where the σ_is are the eigenvalues of the matrix A_1. Now using the transformation of variables $\tilde{W} = T^{-1}W$ we get the symmetrized form of system (3) :

$$\tilde{\mathcal{L}}\tilde{W} := \beta\tilde{W} + B_1\partial_x\tilde{W} + B_2\partial_y\tilde{W} = T^{-1}f \quad , \quad \beta = \frac{1}{\Delta t} \tag{19}$$

In the case of the Euler equations the matrix T is given by :

$$T = \begin{pmatrix} 1 & 0 & 1 & 1 \\ u-c & 0 & u & u+c \\ v & c\sqrt{2} & v & v \\ \frac{1}{2}(u^2+v^2) - cu + \frac{c^2}{\gamma-1} & vc\sqrt{2} & \frac{1}{2}(u^2+v^2) & \frac{1}{2}(u^2+v^2) + cu + \frac{c^2}{\gamma-1} \end{pmatrix}$$

and the matrices of the symmetrized system have the following expressions :

$$B_1 = \mathrm{diag}(u-c\ ,\ u\ ,\ u\ ,\ u+c) \qquad B_2 = \begin{pmatrix} v & \frac{c}{\sqrt{2}} & 0 & 0 \\ \frac{c}{\sqrt{2}} & v & 0 & \frac{c}{\sqrt{2}} \\ 0 & 0 & v & 0 \\ 0 & \frac{c}{\sqrt{2}} & 0 & v \end{pmatrix}$$

In the sequel we simply note W (respectively $\mathcal{L}$) instead of $\tilde{W}$ (respectively $\tilde{\mathcal{L}}$).

3.1.2 The two-subdomain case

We consider the case of a two-subdomain decomposition with $\Omega_1 = \mathbb{R}_- \times \mathbb{R}$, $\Omega_2 = \mathbb{R}_+ \times \mathbb{R}$ separated by the interface $x = 0$; let $\boldsymbol{n} = (1, 0)$ denote the normal vector at the interface $x = 0$, directed from Ω_1 to Ω_2. Let :

$$\begin{cases} M_n &= \dfrac{\mathbf{V}.\boldsymbol{n}}{c} = \dfrac{u}{c} \\ M_t &= \dfrac{\mathbf{V}.\boldsymbol{t}}{c} = \dfrac{v}{c} \end{cases}$$

respectively denote the normal and the tangential Mach number at the interface $x = 0$. We also have that, at any point of $\Omega_1 \bigcup \Omega_2$, the Mach number can be expressed as:

$$M = \frac{\sqrt{u^2+v^2}}{c} = \sqrt{M_u^2 + M_v^2}$$

We denote by $(E_i^p)(x) = (W_i^p - W_i)(x)$ the error vector in the ith subdomain at the pth iteration of the Schwarz algorithm (17). This algorithm, formulated in terms of error vectors, simply writes as :

$$
\begin{array}{c}
\Omega_1 \\
\left\{ \begin{array}{lcl}
\mathcal{L}E_1^{p+1} & = & 0 \text{ for } x < 0 \\
(E_1^{p+1})_j & = & (E_2^p)_j \text{ for } \sigma_j < 0 \ , \ x = 0 \\
E_1^{p+1} & & \text{bounded at } -\infty
\end{array} \right. \\
\Omega_2 \\
\left\{ \begin{array}{lcl}
\mathcal{L}E_2^{p+1} & = & 0 \text{ for } x > 0 \\
(E_2^{p+1})_j & = & (E_1^p)_j \text{ for } \sigma_j > 0 \ , \ x = 0 \\
E_2^{p+1} & & \text{bounded at } +\infty
\end{array} \right.
\end{array}
\tag{20}
$$

where $(E_i^{p+1})_j$ denotes the jth component of the error vector E_i^{p+1}. In order to evaluate the convergence rate of the Schwarz algorithm, we have to solve local boundary value problems. In the present case where we consider the solution of the two-dimensional Euler equations, we cannot do this directly. The mathematical tool which allows us to overcome the difficulty of resolution of the local problems is the Fourier transform. More precisely, we now proceed to a Fourier transform (denoted by $\mathcal{F}$) in the y direction (the Fourier variable is denoted by k). This results in :

$$
\begin{array}{rcl}
\Omega_1 \ : & \left\{ \begin{array}{lcl}
\dfrac{d}{dx}\hat{E}_1^{p+1} & = & -M(k)\hat{E}_1^{p+1} \\
M(k) & = & B_1^{-1}(\beta \mathrm{Id} + ikB_2) \\
(\hat{E}_1^{p+1})_j & = & (\hat{E}_2^p)_j \text{ for } \sigma_j < 0 \ , \ x = 0
\end{array} \right. \\
\Omega_2 \ : & \left\{ \begin{array}{lcl}
\dfrac{d}{dx}\hat{E}_2^{p+1} & = & -M(k)\hat{E}_2^{p+1} \\
M(k) & = & B_1^{-1}(\beta \mathrm{Id} + ikB_2) \\
(\hat{E}_2^{p+1})_j & = & (\hat{E}_1^p)_j \text{ for } \sigma_j > 0 \ , \ x = 0
\end{array} \right.
\end{array}
\tag{21}
$$

We thus obtain local problems that are very simple ODEs whose solutions can be expressed as linear combinations of the eigenvectors of $M(k)$:

$$\hat{E}_i(x,k) = \sum_{j=1}^{4} \alpha_j^i e^{-\lambda_j(k)x} V_j(k) \tag{22}$$

where $\lambda_j(k)$ are the eigenvalues of $M(k)$. This takes place when the eigenvectors are lineraly independent. A very simple calculation leads to the conclusion that the vectors are linealry dependent only when $v = 0$ for $k^2 = c^2/u^2$, in this case the matrix $M(k)$ cannot be diagonalized using eigenvectors. In order to have a general result we should need to tri-diagonalize like in [4]. In order to simplify the calculations we will suppose that $k^2 \neq c^2/u^2$.
We will further require that these solutions are bounded at infinity ($-\infty$ and $+\infty$ respectively) from what we deduce that in the decomposition of $\hat{E}_1(x,k)$ (respectively $\hat{E}_2(x,k)$) we must retain only the eigenvectors corresponding to negative (respectively positive) eigenvalues. In the case of the Euler equations we have :

$$M(k) = \begin{pmatrix} \dfrac{a}{u-c} & \dfrac{ikc}{\sqrt{2}(u-c)} & 0 & 0 \\ \dfrac{ikc}{\sqrt{2}u} & \dfrac{a}{u} & 0 & \dfrac{c}{\sqrt{2}u} \\ 0 & 0 & \dfrac{a}{u} & 0 \\ 0 & \dfrac{c}{\sqrt{2}(u+c)} & 0 & \dfrac{a}{u+c} \end{pmatrix}$$

whith $a = \beta + ikv$. From now, we make the assumption that the flow is subsonic i.e. $M < 1$; this also means that $\dfrac{|u|}{c} < 1$ and $\dfrac{|v|}{c} < 1$ since $M^2 = \dfrac{u^2+v^2}{c}$. Finally we also assume that the flow is such that $u > 0$, in other words we have that $0 < u < c$. Using these hypotheses we obtain the following expressions for the eigenvalues and the corresponding eigenvectors of the matrix $M(k)$:

$$\lambda_1(k) = \frac{-au - R(k)}{c^2 - u^2}, \qquad V_1(k) = \begin{bmatrix} -\dfrac{(R(k)+a)(c+u)}{\sqrt{2}} \\ 0 \\ ik(c^2-u^2) \\ \dfrac{(R(k)-a)(c-u)}{\sqrt{2}} \end{bmatrix}$$

$$\lambda_{2,3}(k) = \frac{a}{u}, \qquad V_2(k) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad V_3(k) = \begin{bmatrix} -\dfrac{iku}{\sqrt{2}} \\ 0 \\ a \\ \dfrac{iku}{\sqrt{2}} \end{bmatrix}$$

$$\lambda_4(k) = \frac{-au + R(k)}{c^2 - u^2}, \qquad V_4(k) = \begin{bmatrix} \dfrac{(R(k)-a)(c+u)}{\sqrt{2}} \\ 0 \\ ik(c^2-u^2) \\ -\dfrac{(R(k)+a)(c-u)}{\sqrt{2}} \end{bmatrix}$$

where $R(k) = \sqrt{a^2 + k^2(c^2-u^2)}$. Under the assumption $0 < u < c$ we have that $\Re(\lambda_1) < 0$ and $\Re(\lambda_{2,3,4}) > 0$ and the the solutions of the local problems (22) become :

$$\begin{cases} \hat{E}_1(x,k) &= \alpha_1 e^{-\lambda_1 x} V_1(k) \\ \hat{E}_2(x,k) &= \alpha_2 e^{-\lambda_2 x} V_2(k) + \alpha_3 e^{-\lambda_3 x} V_3(k) + \alpha_4 e^{-\lambda_4 x} V_4(k) \end{cases} \tag{23}$$

From the interface conditions in (21) we obtain for $x = 0$:

$$\begin{cases} (\hat{E}_1^{p+1})_1 &= (\hat{E}_2^{p})_1 \\ (\hat{E}_2^{p+1})_{2,3,4} &= (\hat{E}_1^{p})_{2,3,4} \end{cases} \tag{24}$$

which is equivalent to :

$$\begin{cases} \alpha_1^{p+1}(V_1(k))_1 = \alpha_2^p(V_2(k))_1 + \alpha_3^p(V_3(k))_1 + \alpha_4^p(V_4(k))_1 \\ \alpha_2^{p+1}(V_2(k))_{2,3,4} + \alpha_3^{p+1}(V_3(k))_{2,3,4} + \alpha_4^{p+1}(V_4(k))_{2,3,4} = \alpha_1^p(V_1(k))_{2,3,4} \end{cases} \tag{25}$$

We deduce from the above relations the following interface iterations :

$$\left\{ \begin{array}{lcl} \alpha_1^{p+1} & = & \mathcal{T}_1 \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}^p = \mathcal{T}_1\mathcal{T}_2\alpha_1^{p-1} \\ \\ \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}^{p+1} & = & \mathcal{T}_2\alpha_1^p = \mathcal{T}_2\mathcal{T}_1 \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}^{p-1} \end{array} \right. \tag{26}$$

which provide the convergence rate of the algorithm :

$$\rho^2_{\text{Schwarz}} = \max\left(\rho(\mathcal{T}_1\mathcal{T}_2)\ ,\ \rho(\mathcal{T}_2\mathcal{T}_1)\right) \tag{27}$$

The Schwarz iterations (25) become :

$$\begin{array}{l} \alpha_1^{p+1} = \mathcal{T}_1 \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}^p = \left[\dfrac{iku}{(R(k)+a)(c+u)}\right]\alpha_3^p - \left[\dfrac{R(k)-a}{R(k)+a}\right]\alpha_4^p \\ \\ \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}^{p+1} = \mathcal{T}_2\alpha_1^p = \begin{pmatrix} 0 \\ \left[\dfrac{2ik(c^2-u^2)(c-u)R(k)}{(R(k)+a)(ac-uR(k))}\right]\alpha_1^p \\ \left[-\dfrac{(R(k)-a)(ac+uR(k))}{(R(k)+a)(ac-uR(k))}\right]\alpha_1^p \end{pmatrix} \end{array} \tag{28}$$

Now we can deduce the convergence rate of the Schwarz algorithm from Eq. (28) :

$$\rho^2_{\text{Schwarz2}}(k) = \left|\frac{R(k)-a}{(R(k)+a)^2}\cdot\frac{R(k)(c-3u)-a(c+u)}{c+u}\right| \tag{29}$$

This quantity depends on 4 parameters : β, k, M_n and M_t i.e. on the time step, frequency, the normal and tangential Mach number at the interface :

$$\left\{ \begin{array}{l} \rho^2_{\text{Schwarz2}}(k) = \left|\dfrac{\tilde{R}(k)-\tilde{a}}{(\tilde{R}(k)+\tilde{a})^2}\cdot\dfrac{\tilde{R}(k)(1-3M_n)-\tilde{a}(1+M_n)}{1+M_n}\right| \\ \tilde{a} = \dfrac{\beta}{c} + ikM_t \\ \tilde{R}(k) = \sqrt{\tilde{a}^2 + k^2(1-M_n^2)} \end{array} \right. \tag{30}$$

One can easily verify that $\rho^2_{\text{Schwarz2}} < 1\ \forall u$, v, c such that M < 1 and $0 < u < c$ and the proposed algorithm is convergent. We also have that in certain conditions the

convergence rate becomes null for a non null value of the frequency. For example, when $M_n < \frac{1}{3}$, $M_t = 0$ and for

$$k^2 = \left(\frac{\beta}{c}\right)^2 \frac{8M_n}{(1-M_n)(1-3M_n)^2}$$

we have $\rho_{Schwarz2}(k) = 0$. Moreover we also have :

$$\begin{aligned} \rho_\infty(M_n, M_t) = \lim_{k\to\infty} \rho_{\text{Schwarz2}}(k) &= \sqrt{1 - \frac{8M_n(1-M_n)(1-M_n^2-M_t^2)}{(1+M_n)^3}} \\ &= \sqrt{\left(\frac{1-3M_n}{1+M_n}\right)^2 + \frac{8M_nM_t^2}{(1+M_n)^3}} \end{aligned} \tag{31}$$

which shows that in the particular case when $M_n^\star = \frac{1}{3}$ and $M_t = 0$ this limit becomes null. Even when $M_t \neq 0$, the convergence rate is minimal for $M_n = M_n^\star$ as we can see on Fig. 1 on which the asympthotic convergence rate is plotted as a function of the normal Mach number at the interface for different values of the tangential Mach number.

This result is surprising and certainly unforseen. For the moment, we retain that this super-convergence behavior is obtained for $M < 1$ and $v = 0$ everywhere in the flowfield which is a rather particular and probably ideal situation. We note in passing that we limit our analysis to a subsonic flow which is the case of interest from the point of view of information propagation at subdomain interfaces. In order to have a better appreciation of the previous result, the behavior of the convergence rate with respect to the frequency number k (which is proportional to h^{-1} i.e. to the number of grid points) is visualized on Fig. 2 to Fig. 5 for different values of M_n and M_t, and for a fixed value of the parameter β i.e. $\beta = \beta_0$. Note that we have only considered situations such that $M_n^2 + M_t^2 < 1$. We remark that the convergence of the Schwarz algorithm deteriorates and becomes less sensible to the value of the normal Mach number as the tangential Mach number tends to 1.

We can also consider different values of the parameter β for a fixed value of M_t in order to get insights on the influence of the time step on the convergence rate (large values of β correspond to small time steps and conversely). The situations where $\beta = 10\beta_0$ and $\beta = 100\beta_0$ for $M_t = 0$ are visualized on Fig. 6. We obtain what we somehow expected i.e. almost the same asymptotic behavior for different values β even though consider larger values of the frequency number have to be considered as β is increased.

We conclude this section by the following remark.

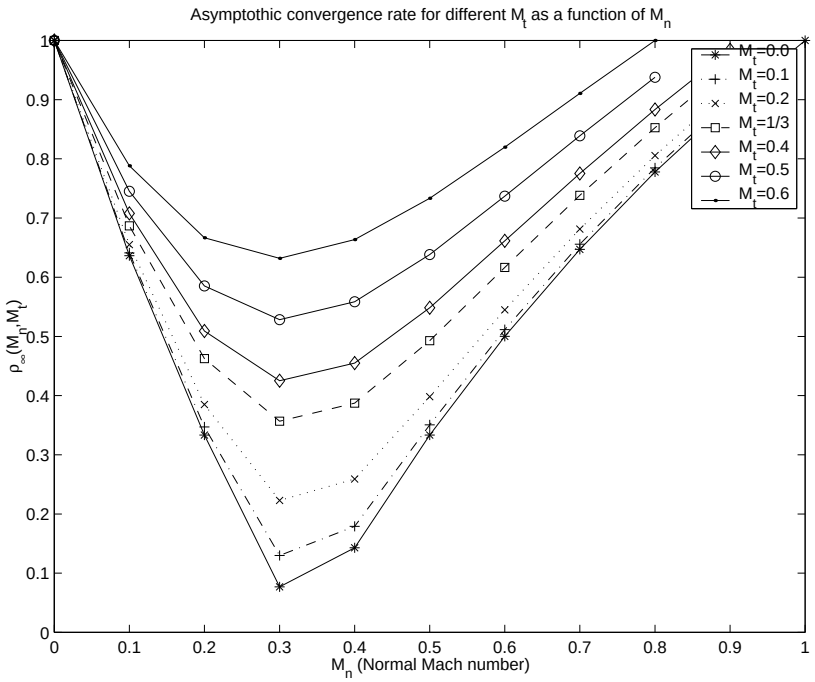

Figure 1: Asympthotic convergence rate of the Schwarz algorithm

Remark.. *When we deal with a larger number of subdomains, we cannont evaluate easily the convergence rate using the approach adopted so far since this translates into the evaluation of the spectral radius of a* $4(N-1) \times 4(N-1)$ *matrix. Estimations under the form of inequalities[15] are possible and this could be a logical continuation of the present work. Here, this aspect has only been investigated experimentally (see the results section) showing that the convergence deteriorates as the number of subdomains is increased even though the dependence on the normal mach number is qualitatively the same than in the two-subdomain case.*

3.2 The three-dimensional case

The objective of this section is to assess if and how the convergence result given by Eq. (30) is affected when the space dimension is increased and when we start from a different formulation of the Euler equations. For this purpose, we study the convergence of the proposed Schwarz algorithm when applied to the solution of the three-dimensional Euler equations expressed in primitive variables. As in the previous section, we limit ourselves to the two-subdomain case. The quasi-linear form of these equations is given by :

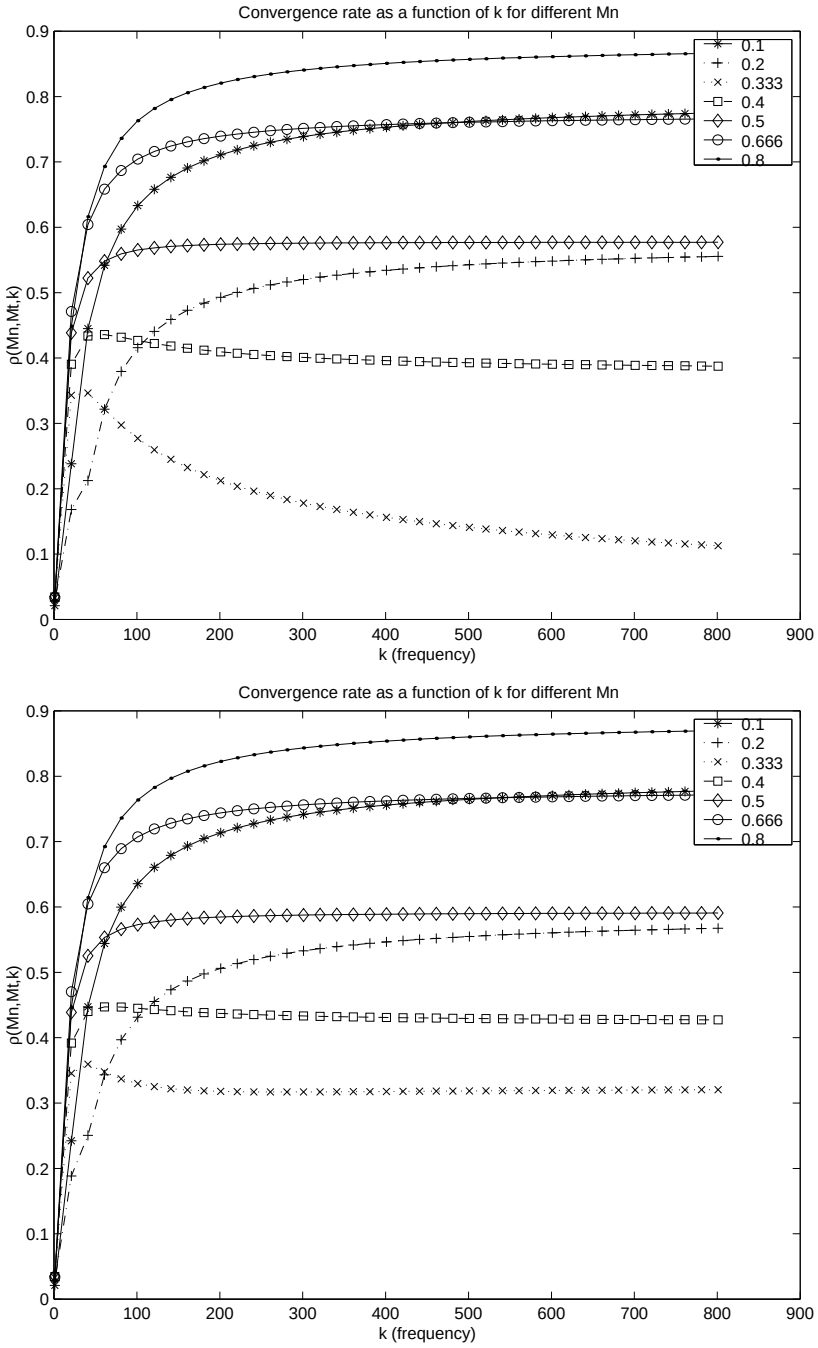

Figure 2: Convergence rate of the Schwarz algorithm in the 2 subdomain case
Top figure : $M_t = 0$ - Bottom figure : $M_t = 0.1$

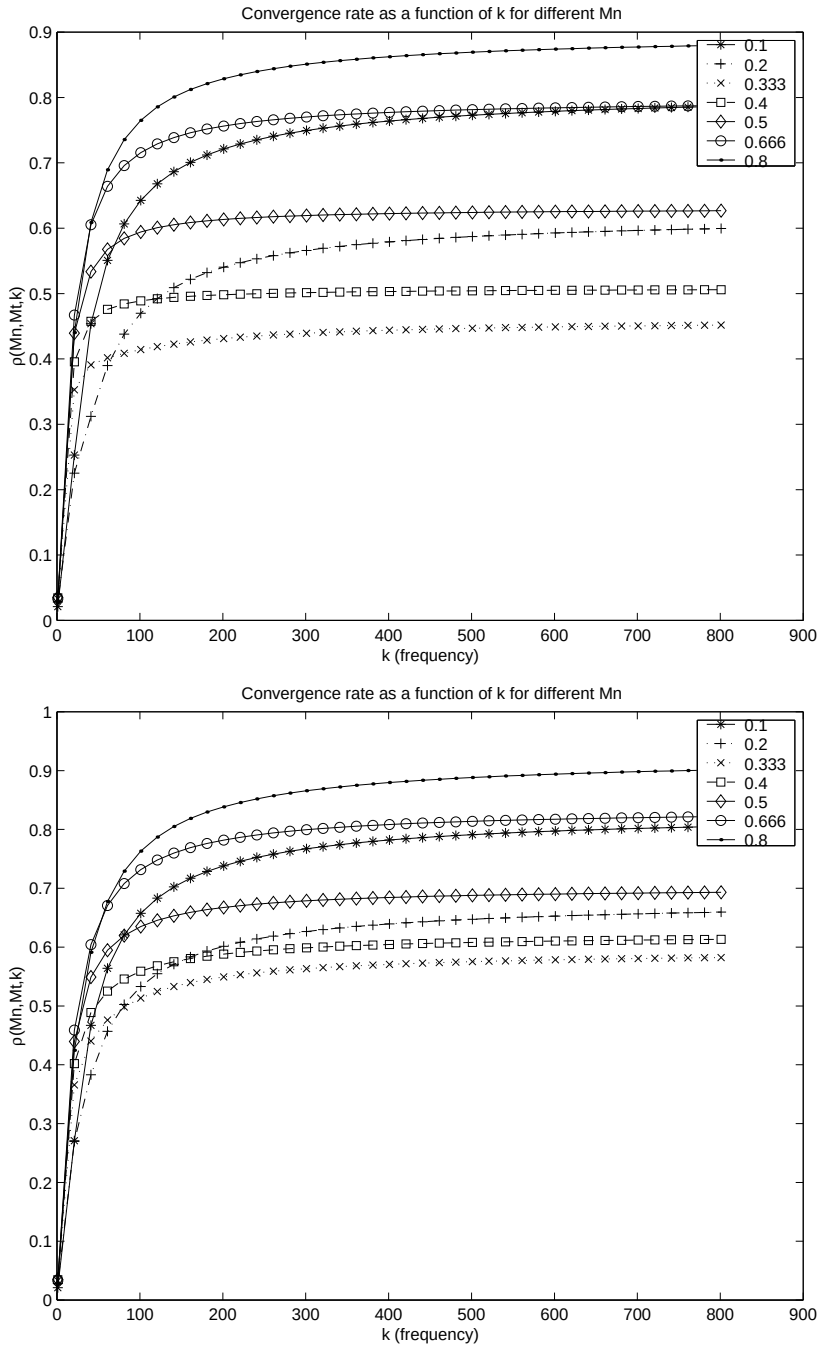

Figure 3: Convergence rate of the Schwarz algorithm in the 2 subdomain case
Top figure : $M_t = 0.2$ - Bottom figure : $M_t = \frac{1}{3}$

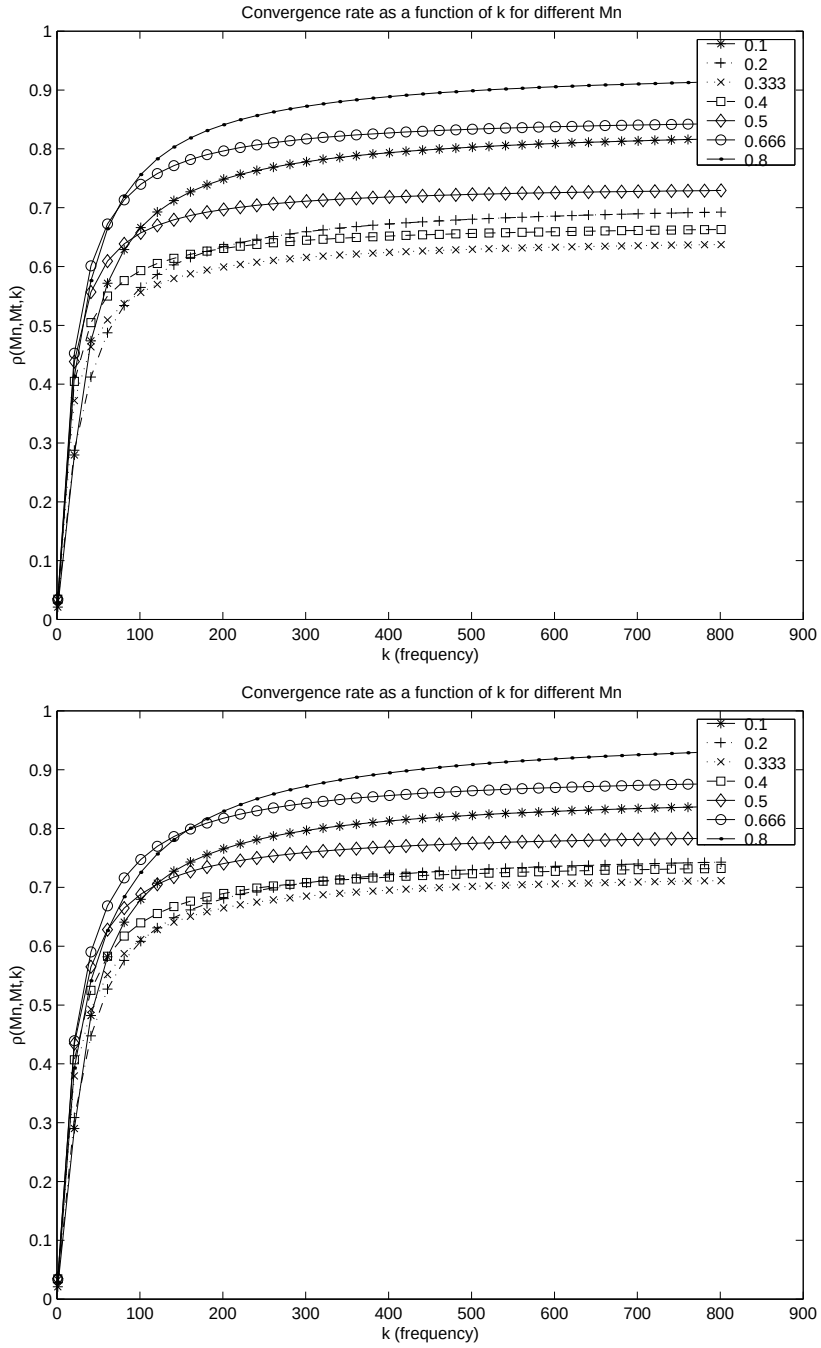

Figure 4: Convergence rate of the Schwarz algorithm in the 2 subdomain case
Top figure : $M_t = 0.4$ - Bottom figure : $M_t = 0.5$

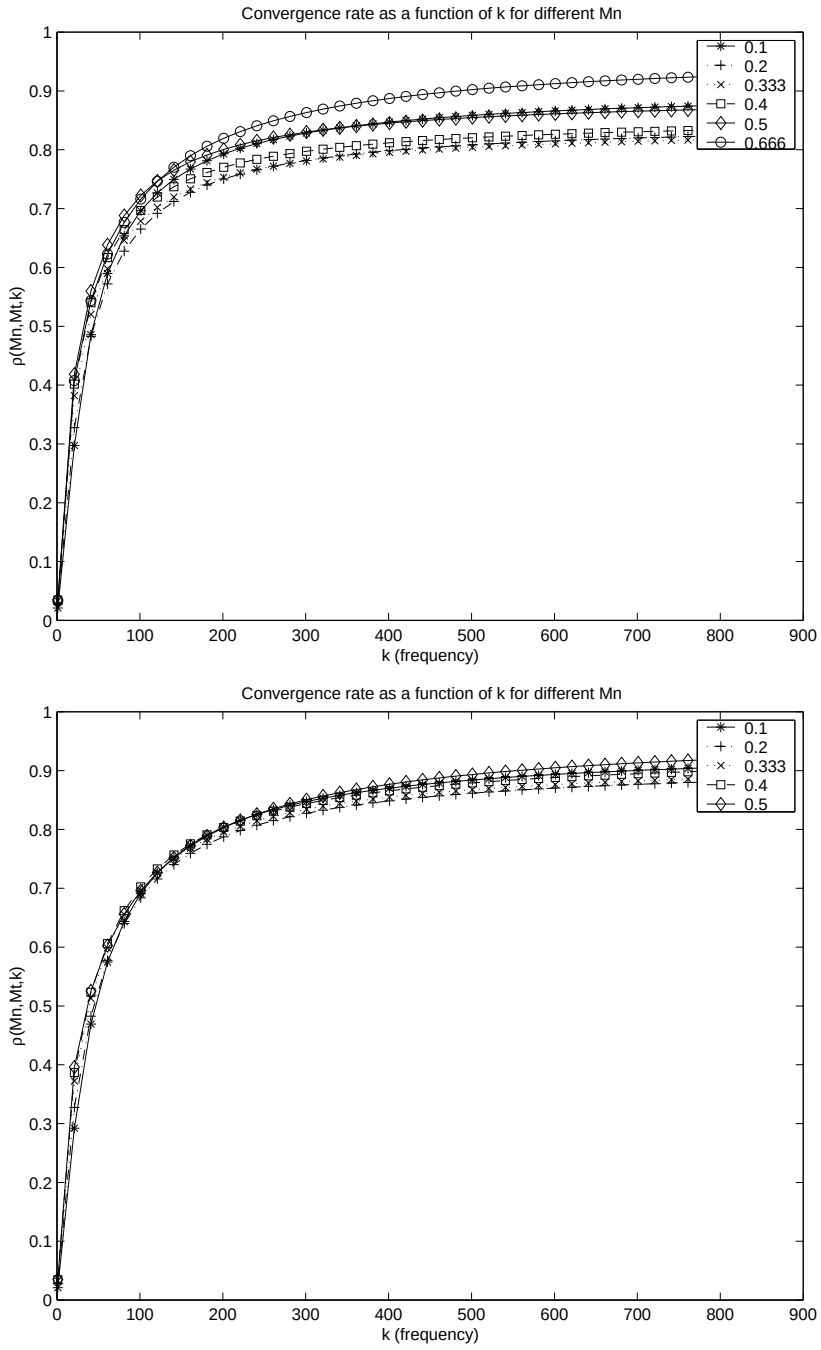

Figure 5: Convergence rate of the Schwarz algorithm in the 2 subdomain case
Top figure : $M_t = \dfrac{1}{6}$ - Bottom figure : $M_t = 0.8$

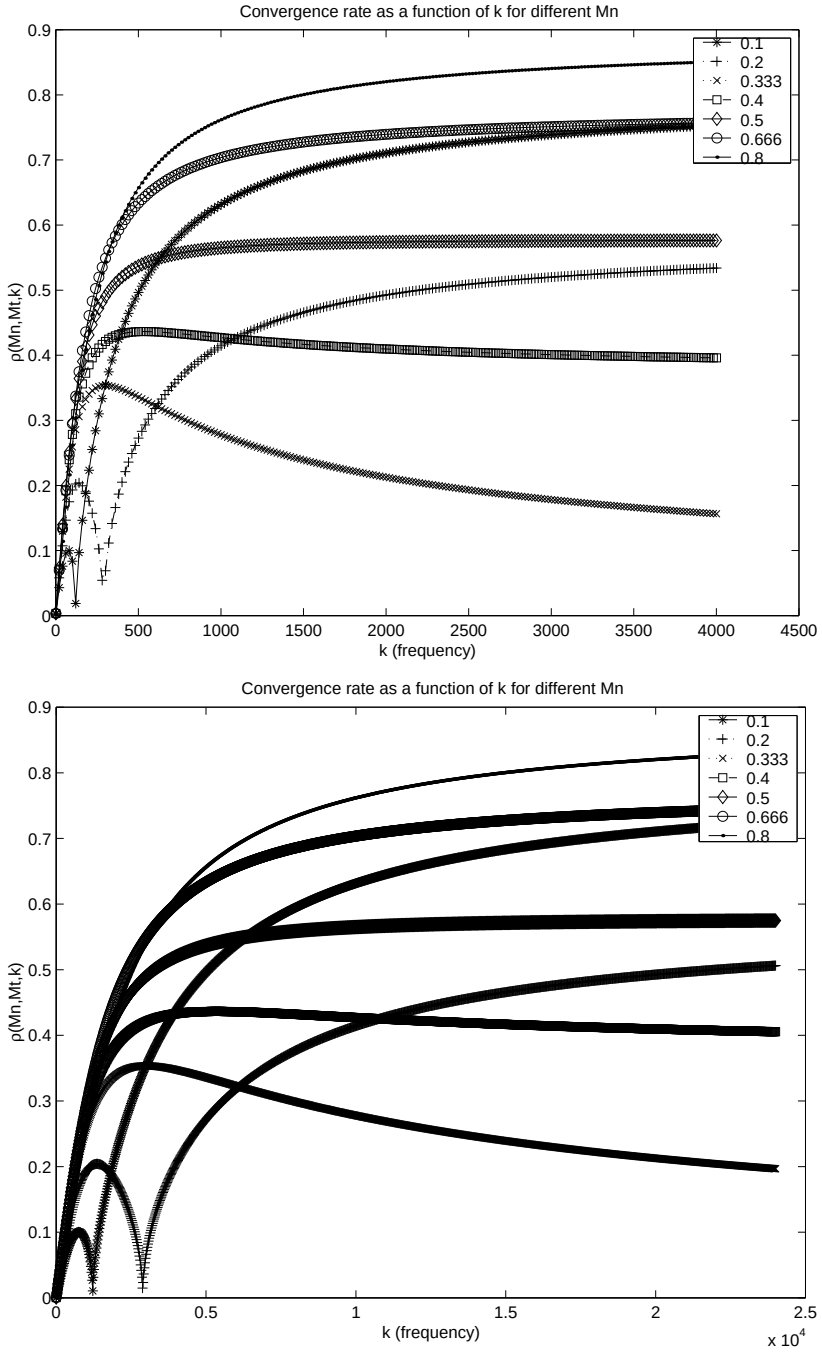

Figure 6: Convergence rate of the Schwarz algorithm for $M_t = 0$: $\beta = 10\beta_0$ (top figure) $\beta = 100\beta_0$ (bottom figure)

$$\partial_t W + A_1(W)\partial_x W + A_2(W)\partial_y W + A_3(W)\partial_z W = 0 \tag{32}$$

where the unknown vector is $W = (\rho\ ,\ u\ ,\ v\ ,\ w\ ,\ p)^T$; the Jacobian matrices of the flux vectors $A_1(W) = \dfrac{\partial F_1(W)}{\partial W}$, $A_2(W) = \dfrac{\partial F_2(W)}{\partial W}$ and $A_3(W) = \dfrac{\partial F_3(W)}{\partial W}$ being given by :

$$A_1(W) = \begin{pmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & \dfrac{1}{\rho} \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \rho\, c^2 & 0 & 0 & u \end{pmatrix} \quad A_2(W) = \begin{pmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & \dfrac{1}{\rho} \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \rho\, c^2 & 0 & v \end{pmatrix}$$

$$A_3(W) = \begin{pmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & \dfrac{1}{\rho} \\ 0 & 0 & 0 & \rho\, c^2 & w \end{pmatrix}$$

As previously, the starting-point of the convergence analysis is given by the linearized system :

$$\mathcal{L}W := \frac{1}{\Delta t} W + A_1\partial_x W + A_2\partial_y W + A_3\partial_z W + = f \tag{33}$$

As in the two-dimensional case, we proceed to the variable change $\tilde{W} = T^{-1}W$ where :

$$T = \begin{pmatrix} -\dfrac{\rho}{c} & 0 & 1 & 0 & \dfrac{\rho}{c} \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -c\,\rho & 0 & 0 & 0 & c\,\rho \end{pmatrix}$$

is such that $A_1 = T\Lambda T^{-1}$ (i.e. T is the matrix whose columns are eigenvectors of A_1) and :

$$\Lambda = \begin{pmatrix} u-c & 0 & 0 & 0 & 0 \\ 0 & u & 0 & 0 & 0 \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & u+c \end{pmatrix}$$

is the diagonal matrix of the corresponding eigenvalues. System (33) becomes :

$$\tilde{\mathcal{L}}\tilde{W} := \beta\tilde{W} + B_1\partial_x\tilde{W} + B_2\partial_y\tilde{W} + B_3\partial_z\tilde{W} = T^{-1}f \quad , \quad \beta = \frac{1}{\Delta t} \tag{34}$$

where $B_2 = T^{-1}A_2T$ and $B_3 = T^{-1}A_3T$:

$$B_2 = \begin{pmatrix} v & -\frac{1}{2}c & 0 & 0 & 0 \\ -c & v & 0 & 0 & c \\ 0 & 0 & v & 0 & 0 \\ 0 & 0 & 0 & v & 0 \\ 0 & \frac{1}{2}c & 0 & 0 & v \end{pmatrix} \quad B_3 = \begin{pmatrix} w & 0 & 0 & -\frac{1}{2}c & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ -c & 0 & 0 & w & c \\ 0 & 0 & 0 & \frac{1}{2}c & w \end{pmatrix}$$

We consider a decomposition in two subdomains where the interface is given by the plane $x = 0$. The Schwarz algorithm is formulated as in (20). We now proceed to a Fourier transform (denoted by $\mathcal{F}$) in the y and z direction (the Fourier variables are denoted by k_1 and k_2). In each subdomain the resulting ODE takes the form :

$$\frac{d}{dx}\hat{E}_i^{p+1}(x,k_1,k_2) = -M(k_1,k_2)\hat{E}_i^{p+1}(x,k_1,k_2)$$

where the matrix $M(k_1,k_2)$ is given by :

$$\begin{pmatrix} -\frac{\tilde{\beta}}{c-u} & \frac{1}{2}\frac{ik_1c}{c-u} & 0 & \frac{1}{2}\frac{ik_2c}{c-u} & 0 \\ -\frac{ik_1c}{u} & \frac{\tilde{\beta}}{u} & 0 & 0 & \frac{ik_1c}{u} \\ 0 & 0 & \frac{\tilde{\beta}}{u} & 0 & 0 \\ -\frac{ik_2c}{u} & 0 & 0 & \frac{\tilde{\beta}}{u} & \frac{ik_2c}{u} \\ 0 & \frac{1}{2}\frac{ik_1c}{u+c} & 0 & \frac{1}{2}\frac{ik_2c}{u+c} & \frac{\tilde{\beta}}{u+c} \end{pmatrix}$$

with $\tilde{\beta} = \beta + ik_1v + ik_2w$. The solution in each subdomain is expressed as a linear combination of the eigenvectors of $M(k_1, k_2)$. From now, we make the assumption that the flow is subsonic i.e. $M < 1$; we also assume $0 < u < c$. Using these hypotheses we obtain the following expressions for the eigenvalues and the corresponding eigenvectors of the matrix $M(k_1, k_2)$:

$$V^-(k_1,k_2) = \begin{bmatrix} \frac{i(R(k_1,k_2)-a)}{2k_2(c-u)} \\ \frac{k_1}{k_2} \\ 0 \\ 1 \\ \frac{i(R(k_1,k_2)+a)}{2k_2(c-u)} \end{bmatrix} \quad V_1^+(k_1,k_2) = \begin{bmatrix} -\frac{i(R(k_1,k_2)+a)}{2k_2(c-u)} \\ \frac{k_1}{k_2} \\ 0 \\ 1 \\ -\frac{i(R(k_1,k_2)-a)}{2k_2(c-u)} \end{bmatrix}$$

$$V_2^+(k_1,k_2) = \begin{bmatrix} \frac{ik_1u}{2a} \\ 1 \\ 0 \\ 0 \\ \frac{ik_1u}{2a} \end{bmatrix} \quad V_3^+(k_1,k_2) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad V_4^+(k_1,k_2) = \begin{bmatrix} \frac{ik_2u}{2a} \\ 1 \\ 0 \\ 0 \\ \frac{ik_2u}{2a} \end{bmatrix}$$

where $a = \beta + ik_1v + ik_2w$ and $R(k_1, k_2) = \sqrt{a^2 + (k_1^2 + k_2^2)(c^2 - u^2)}$. The associated eigenvalues are :

$$\begin{cases} \lambda_1(k_1,k_2) &= \dfrac{-au - cR(k_1,k_2)}{c^2-u^2} \\ \lambda_2(k_1,k_2) &= \dfrac{-au + cR(k_1,k_2)}{c^2-u^2} \\ \lambda_{3,4,5}(k_1,k_2) &= \dfrac{a}{u} \end{cases}$$

Under the assumption $0 < u < c$ we have that $\Re(\lambda_1) < 0$ and $\Re(\lambda_{2,3,4,5}) > 0$. The Schwarz iteration at the interface can be expressed for the first and the second subdomain as :

$$\begin{aligned} \alpha_1^{p+1}(V^-(k_1,k_2))_1 &= \alpha_2^p(V_1^+(k_1,k_2))_1 + \alpha_3^p(V_2^+(k_1,k_2))_1 \\ &+ \alpha_4^p(V_3^+(k_1,k_2))_1 + \alpha_5^p(V_4^+(k_1,k_2))_1 \end{aligned}$$

and :

$$\left\{ \begin{array}{rcl} \alpha_2^{p+1}(V_1^+(k_1,k_2))_2 + \alpha_3^{p+1}(V_2^+(k_1,k_2))_2 + \alpha_4^{p+1}(V_3^+(k_1,k_2))_2 + \alpha_5^{p+1}(V_4^+(k_1,k_2))_2 & = & \alpha_1^p(V^-(k_1,k_2))_2 \\ \alpha_2^{p+1}(V_1^+(k_1,k_2))_3 + \alpha_3^{p+1}(V_2^+(k_1,k_2))_3 + \alpha_4^{p+1}(V_3^+(k_1,k_2))_3 + \alpha_5^{p+1}(V_4^+(k_1,k_2))_3 & = & \alpha_1^p(V^-(k_1,k_2))_3 \\ \alpha_3^{p+1}(V_1^+(k_1,k_2))_4 + \alpha_3^{p+1}(V_2^+(k_1,k_2))_4 + \alpha_4^{p+1}(V_3^+(k_1,k_2))_4 + \alpha_5^{p+1}(V_4^+(k_1,k_2))_4 & = & \alpha_1^p(V^-(k_1,k_2))_4 \\ \alpha_3^{p+1}(V_1^+(k_1,k_2))_5 + \alpha_3^{p+1}(V_2^+(k_1,k_2))_5 + \alpha_4^{p+1}(V_3^+(k_1,k_2))_5 + \alpha_5^{p+1}(V_4^+(k_1,k_2))_5 & = & \alpha_1^p(V^-(k_1,k_2))_5 \end{array} \right.$$

where the indices of the vectors denote the vector components. By replacing the expressions of the eigenvectors in the above equations and solving the resulting equations for the α_i we get :

$$\alpha_1^{p+1} = -\left[\frac{R(k_1,k_2)-a}{R(k_1,k_2)-a}\right]\alpha_2^p - \left[\frac{k_1k_2u(c-u)}{a(R(k_1,k_2)+a)}\right]\alpha_3^p - \left[\frac{k_2^2u(c-u)}{a(R(k_1,k_2)+a)}\right]\alpha_5^p$$

and :

$$\left\{ \begin{array}{rcl} \alpha_2^{p+1} &=& -\left[\frac{R(k_1,k_2)-a}{R(k_1,k_2)+a}\right]\cdot\left[\frac{ac+uR(k_1,k_2)}{ac-uR(k_1,k_2)}\right]\alpha_1^p \\ \alpha_3^{p+1} &=& \left[\frac{2ak_1(c-u)R(k_1,k_2)}{k_2(R(k_1,k_2)+a)(ac-uR(k_1,k_2))}\right]\alpha_1^p \\ \alpha_4^{p+1} &=& 0 \\ \alpha_5^{p+1} &=& \left[\frac{2a(c-u)R(k_1,k_2)}{(R(k_1,k_2)+a)(ac-uR(k_1,k_2))}\right]\alpha_1^p \end{array} \right.$$

Finally, the convergence rate of the above algorithm is :

$$\rho^2_{\text{Schwarz2}}(k_1,k_2) = \left| \left[\frac{R(k_1,k_2)-a}{R(k_1,k_2)+a}\right]^2 - \left[\frac{4u(R(k_1,k_2)-a)}{(R(k_1,k_2)+a)^2(c+u)}\right] \right|$$

As we can see the convergence rate in the $3D$ case has a similar form to one in the $2D$ case, but depending on the 2 wavenumbers $k_{1,2}$. Therefore, its asympthotic behaviour as these 2 parameters tend to ∞ and when the Mach numbers in the directions x and y will be null is identical to that of the $2D$ case. As a conclusion, qualitatively, we have the same result irrespective of the space dimension.

4 Numerical results

4.1 Main characteristics of the flow solver

Here, we briefly review the main characteristics of the flow solver which is described in details in Dolean and Lanteri[3].

4.1.1 Starting-point flow solver

The Euler equations are solved in conservative form. The flow domain Ω is discretized by a triangulation $\mathcal{T}_h$ where h is the maximal length of the edges of $\mathcal{T}_h$. A vertex of $\mathcal{T}_h$ is denoted by s_i and the set of neighboring vertices of s_i by $N(i)$. We associate to each vertex s_i a control surface (or cell) denoted by C_i which is constructed as the union of local contributions from the set of triangles sharing s_i. The contribution of a given triangle is obtained by joining its barycenter G to the midpoints I of the edges incident to s_i (see Fig. 7). The boundary of C_i is denoted by ∂C_i and the unitary normal vector exterior to ∂C_i by $\vec{\nu}_i = (\nu_{ix}, \nu_{iy})$. The union of all these cells constitutes a discretization of Ω often qualified as dual to $\mathcal{T}_h$:

$$\Omega_h = \bigcup_{i=1}^{N_V} C_i \quad , \quad N_V : \text{number of vertices of } \mathcal{T}_h$$

The spatial discretization method combines the following ingredients :

- a finite volume formulation together with an upwind scheme for the discretization of the convective flux. In this study, the numerical flux function corresponds to the approximate Riemann solver of Roe[19];

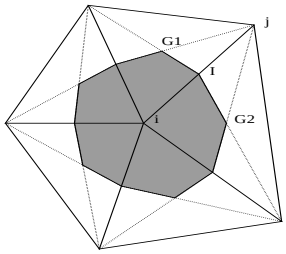

Figure 7: A control surface on a triangular mesh

- extension to second order accuracy is obtained by using the MUSCL (Monotonic Upstream Schemes for Conservation Laws) which was introduced by van Leer[21] and extended to unstructured triangular meshes by Fezoui and Stoufflet[6].

Time integration of the resulting semi-discrete equations relies on the implicit linearized formulation described in Fezoui and Stoufflet[6]. Then, at each pseudo-time step, a linear system must be solved to advance the solution in time. This is where the domain decomposition approach proposed in section 2 is introduced.

4.1.2 Domain decomposition solver

The flow solver is parallelized using a classical strategy that combines domain partitioning techniques and a message-passing programming model[5]. Here, according to the domain decomposition algorithm formulated in section 2, it is interesting to consider mesh partitions involving a one-triangle wide overlapping region which is shared by neighboring subdomains. As a matter of fact, it is easily seen that within this setting, the interface between two neighboring subdomains is a non-overlapping one from the viewpoint of the dual discretization of Ω in terms of control surfaces; if Ω_1 and Ω_2 are neighbors then :

$$\Gamma = \Omega_1 \cap \Omega_2 = \bigcup_{C_{1_k} \in \Omega_1, C_{2_k} \in \Omega_2} \partial C_{1_k} \cap \partial C_{2_k}$$

In order to be able to construct an interface system, we need to consider a preliminary step which consists in the introduction of a redundant variable at the interface between two control surfaces (see Fig. 8), following a strategy adopted by Clerc[2]. Our approach is however different from the one described in [2] when considering the nature of this redundant variable; as detailed below, the latter is defined as the normal flux between two control surfaces belonging to different subdomains.

To simplify the presentation we consider the case of a decomposition of Ω in two subdomains. Let $[s_i, s_j]$ be and edge such that C_i (associated with s_i) and C_j (associated with s_j) belong to two neighboring subdomains. An additive Schwarz formulation is obtained by setting the following interface conditions :

$$\left\{ \begin{array}{rcl} \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) W_i^{(k+1)} & = & \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) W_j^{(k)} \\ \mathcal{A}^+_{\vec{\nu}_{ij}}(\tilde{W}^n) W_j^{(k+1)} & = & \mathcal{A}^+_{\vec{\nu}_{ij}}(\tilde{W}^n) W_i^{(k)} \end{array} \right. \tag{35}$$

where $\tilde{W}^n$ is the mean value or Roe[19] of the state vectors W_i and W_j and where we have noted $\mathcal{A}^{\pm}_{\vec{\nu}_{ij}}(\tilde{W}^n) = \mathcal{A}^{\pm}_R(W_i, W_j, \vec{\nu}_{ij})$ with :

$$\begin{array}{rcl} \mathcal{A}_{\vec{\nu}}(W) & = & \nu_x A_1(W) \quad + \quad \nu_y A_2(W) \\ & = & \nu_x \dfrac{\partial F_1(W)}{\partial W} \quad + \quad \nu_y \dfrac{\partial F_2(W)}{\partial W} \end{array}$$

Conditions (35) are expressing the continuity of normal fluxes at the subdomain interface $\Gamma = \Omega_1 \cap \Omega_2$.

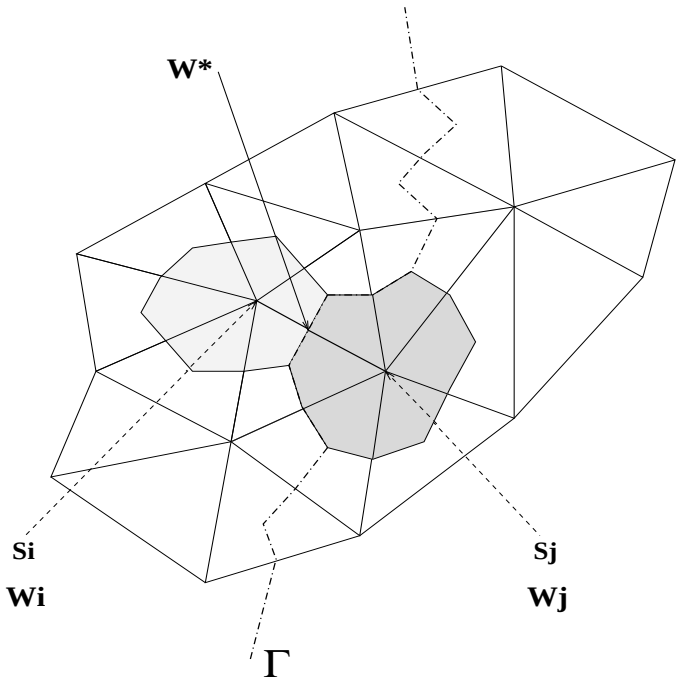

Figure 8: Definition of a redundant variable at an interface $\Gamma = \Omega_1 \cap \Omega_2$

In the sequel, we simply write W_i instead of $W_i^{(k+1)}$. We introduce an auxiliary variable denoted by W^* and such that :

$$\begin{aligned} \left(\mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n)W_j\right)|_{s_j} &= \left(\mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n)W^*\right)|_{\frac{s_i+s_j}{2}} \\ &\text{and} \\ \left(\mathcal{A}^+_{\vec{\nu}_{ij}}(\tilde{W}^n)W_i\right)|_{s_i} &= \left(\mathcal{A}^+_{\vec{\nu}_{ij}}(\tilde{W}^n)W^*\right)|_{\frac{s_i+s_j}{2}} \end{aligned} \tag{36}$$

and we define :

$$\Phi = |\mathcal{A}_{\vec{\nu}_{ij}}(\tilde{W}^n)|W^* = \mathcal{A}^+_{\vec{\nu}_{ij}}(\tilde{W}^n)W_i - \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n)W_j \tag{37}$$

the associated new unknown of the problem. We can write :

$$\Phi = \left(\mathcal{T}(\tilde{W}^n)|\Lambda(\tilde{W}^n)|\mathcal{T}^{-1}(\tilde{W}^n)\right)W^* \quad \Leftrightarrow \quad W^* = \left(\mathcal{T}(\tilde{W}^n)|\Lambda(\tilde{W}^n)|^{-1}\mathcal{T}^{-1}(\tilde{W}^n)\right)\Phi$$

where $\Lambda(\tilde{W}^n)$ is the diagonal matrix whose components are the eigenvalues of $\mathcal{A}_{\vec{\nu}_{ij}}(\tilde{W}^n)$ and $\mathcal{T}(\tilde{W}^n)$ is the matrix whose columns are the associated left eigenvectors. The positive and negative parts of this flux are given by :

$$\begin{aligned} \Phi^{\pm} &= \mathcal{A}^{\pm}_{\vec{\nu}_{ij}}(\tilde{W}^n)W^* \\ &= \left(\mathcal{T}(\tilde{W}^n)\Lambda^{\pm}(\tilde{W}^n)\mathcal{T}^{-1}(\tilde{W}^n)\right)W^* \\ &= \left(\mathcal{T}(\tilde{W}^n)\Lambda^{\pm}(\tilde{W}^n)|\Lambda^{-1}(\tilde{W}^n)|\mathcal{T}^{-1}(\tilde{W}^n)\right)\Phi \end{aligned} \tag{38}$$

that we write in condensed form as :

$$\Phi^{\pm} = P^{\pm}(\tilde{W}^n)\Phi$$

On the other hand, the elementary fluxes associated with the control surfaces C_i and C_j are computed using the numerical flux function characterizing the approximate Riemann solver of Roe. As a consequence, they can be expressed in terms of the auxiliary flux Φ as (see [3] for more details) :

$$\begin{cases} \Phi(W_i, W_j, \vec{\nu}_{ij}) &= (\mathcal{A}_{\vec{\nu}_{ij}}(W_i^n) - \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n))W_i + \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n)W_j \\ \Phi(W_j, W_i, \vec{\nu}_{ji}) &= -\mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n)W_j - (\mathcal{A}_{\vec{\nu}_{ij}}(W_i^n) - \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n))W_i \end{cases}$$

On the other hand, the elementary linearized fluxes associated with the control surfaces C_i and C_j can be written as :

$$\left\{ \begin{array}{rcl} \Phi_c(W_i, W_j, \vec{\nu}_{ij}) & = & \left(\mathcal{A}_{\vec{\nu}_{ij}}(W_i^n) - \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) \right) W_i + \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) W_j \\ \Phi_c(W_j, W_i, \vec{\nu}_{ji}) & = & -\mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) W_j - \left(\mathcal{A}_{\vec{\nu}_{ij}}(W_i^n) - \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) \right) W_i \end{array} \right.$$

By making the following approximation at the interface :

$$\mathcal{A}_{\vec{\nu}_{ij}}(W_i^n) - \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) \cong \mathcal{A}^+_{\vec{\nu}_{ij}}(\tilde{W}^n)$$

we can further use the relations (36), (37) and (38) to get the expression of the interface flux using the new variable Φ_c :

$$\left\{ \begin{array}{rcl} \Phi_c^i(W_i, W^\star, \vec{\nu}_{ij}) & = & \left(\mathcal{A}_{\vec{\nu}_{ij}}(W_i^n) - \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) \right) W_i + \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) W^* \\ & = & \left(\mathcal{A}_{\vec{\nu}_{ij}}(W_i^n) - \mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) \right) W_i + P^-(\tilde{W}^n) \Phi_c \\ \Phi_c^j(W^\star, W_j, \vec{\nu}_{ij}) & = & -\mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) W_j - \mathcal{A}^+_{\vec{\nu}_{ij}}(\tilde{W}^n) W^* \\ & = & -\mathcal{A}^-_{\vec{\nu}_{ij}}(\tilde{W}^n) W_j - P^+(\tilde{W}^n) \Phi_c \end{array} \right. \tag{39}$$

Taking into account Eq. (37) and (39) we can construct an implicit linear system that distinguishes purely interior unknowns (state vectors) from interface ones (normal fluxes) :

$$\left(\begin{array}{ccc} \mathcal{M}_1 & 0 & \mathcal{M}_{1\Phi} \\ 0 & \mathcal{M}_2 & \mathcal{M}_{2\Phi} \\ \mathcal{F}_1 & \mathcal{F}_2 & \mathrm{I}d \end{array} \right) \left(\begin{array}{c} W_1 \\ W_2 \\ \Phi \end{array} \right) = \left(\begin{array}{c} b_1 \\ b_2 \\ 0 \end{array} \right) \tag{40}$$

where $\mathcal{M}_1$ (respectively $\mathcal{M}_2$) is the matrix that couples the unknowns associated with vertices internal to Ω_1 (respectively Ω_2) whereas $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{M}_{1\Phi}$ and $\mathcal{M}_{2\Phi}$ are coupling matrices between internal and interface unknowns. These various matrix terms are detailed in [3]. Now, the internal unknowns can be eliminated in favor of the interface ones to yield the following interface system :

$$\begin{array}{rcl} S\Phi & = & [\mathrm{I}d - (\mathcal{F}_1 \mathcal{M}_1^{-1} \mathcal{M}_{1\Phi} + \mathcal{F}_2 \mathcal{M}_2^{-1} \mathcal{M}_{2\Phi})]\Phi \\ & = & g \\ & = & -[\mathcal{F}_1 \mathcal{M}_1^{-1} b_1 + \mathcal{F}_2 \mathcal{M}_2^{-1} b_2] \end{array} \tag{41}$$

As usual in this context, once this system has been solved for Φ, we obtain the values of the purely internal unknowns by performing independent (i.e. parallel) local solves :

$$\left\{ \begin{array}{lcl} W_1 & = & \mathcal{M}_1^{-1}(b_1 - \mathcal{M}_{1\Phi}\Phi) \\ W_2 & = & \mathcal{M}_2^{-1}(b_2 - \mathcal{M}_{2\Phi}\Phi) \end{array} \right. \tag{42}$$

4.1.3 Interface solvers

A simple algorithm for solving system (41) is given by the following Richardson type iteration.

ALGORITHM 1 *(***RICH***) : Richardson type iteration for solving the interface system* $S\Phi = g$.

- *Initialisation :* $\Phi = \Phi^\circ$
- *Computation of* $g = g_1 + g_2$ *(including communication steps to assemble local contributions) with :*
  $g_i = \mathcal{F}_i x_i$ *where* x_i *is obtained through the local solution :* $\mathcal{M}_i x_i = b_i$
- *Main parallel loop :* $k = 0, .., K$
  - *Subdomain* Ω_1 *:* $y_1 = \mathcal{M}_{12}\Phi^k$ *and* $\Phi_1 = \mathcal{F}_1 v_1$
    where v_1 *is obtained through the local solution :* $\mathcal{M}_1 v_1 = y_1$
  - *Subdomain* Ω_2 *:* $y_2 = \mathcal{M}_{21}\Phi^k$ *and* $\Phi_2 = \mathcal{F}_2 v_2$
    where v_2 *is obtained through the local solution :* $\mathcal{M}_2 v_2 = y_2$
  - *Assembly process (including communication steps) :* $\Phi^{k+1} = \Phi_1 + \Phi_2 + g$
- *If* $\|\Phi^{k+1} - \Phi^k\| < \epsilon$ *then exit main loop*

In order to solve the linear system $S\Phi = g$ (see Eq. (41)) we have considered the following three strategies :

- a Richardson type iteration (i.e. algorithm 1);
- a full GMRES iteration[20].

4.1.4 Local solution strategies

The domain decomposition algorithm proposed in subsection 4.1.2 calls for independent (parallel) local solution steps in each subdomain. Here, we are interested in solving the corresponding linear systems iteratively using a multigrid strategy. A Gauss-Seidel method is used as a smoother and the multigrid method is the accelerator. The multigrid method considered in the present study aims at accelerating the iterative solution of linear systems resulting from the adoption of a linearized implicit scheme for time advancing the equations modelling compressible fluid flows. It is well known that classical relaxation methods such as the Jacobi or Gauss-Seidel methods, applied to the iterative solution of the resulting linear system, quickly damp the high frequencies of the error however they do not allow for an efficient treatment of the low frequency components. The basic idea of the coarse grid correction scheme is to transfer the partially solved solution on a coarser grid in order to transform the low frequencies of the fine grid solution in high frequencies which are then efficiently damped by the standard relaxation methods. The method is based on a grid coarsening by agglomeration technique for the construction of coarse hrid levels; the method is described in details in [13].

4.2 Test cases definition

The numerical simulations considered here aim at assessing the convergence results of section 3 from an experimental point of view. In order to do so we concentrate on the solution of the linear system resulting from the first implicit time step starting from a uniform flow. If not explicitly stated otherwise, the linear thresholds for the local (ε_l) and the interface (ε_i) system solutions are given by $\varepsilon_l = \varepsilon_i = 10^{-10}$. On the other hand, the CFL number is set to the value 1000 for all the numerical simulations. For these numerical experiments, we have considered two geometries : a rectangular domain of size $[0,8] \times [0,1]$ and a `NACA0012` airfoil. For the first geometry, two types of discretization have been used : the first type consists in a regular triangulation (i.e. obtained from a finite difference grid, see Fig. 9) while the second type is an unstructured triangulation (see Fig. 10). The characteristics of the unstructured triangulations are given in Tab. 2. Meshes RS2 and RS3 (respectively, meshes RU2 and RU3) have been obtained by uniform divisions of mesh RS1 (respectively, mesh RU1). For both geometries, the initialisation is given by the uniform flow characterised by $\rho_0 = 1$, $u_0 = 1$, $v_0 = 0$ and $p_0 = \dfrac{1}{\gamma M^2}$ where M denotes the freestream Mach number. For the first geometry, a slip condition is applied on the horizontal sides while an inflow (respectively, outflow) condition is applied on the left (respectively, right) vertical side. Concerning the `NACA0012` airfoil, three unstructured triangular meshes have been used

whose characteristics are given in Tab. 3 (see Fig. 11 for a partial view of mesh N1). Mesh N2 and N3 have been obtained by uniform divisions of mesh N1.

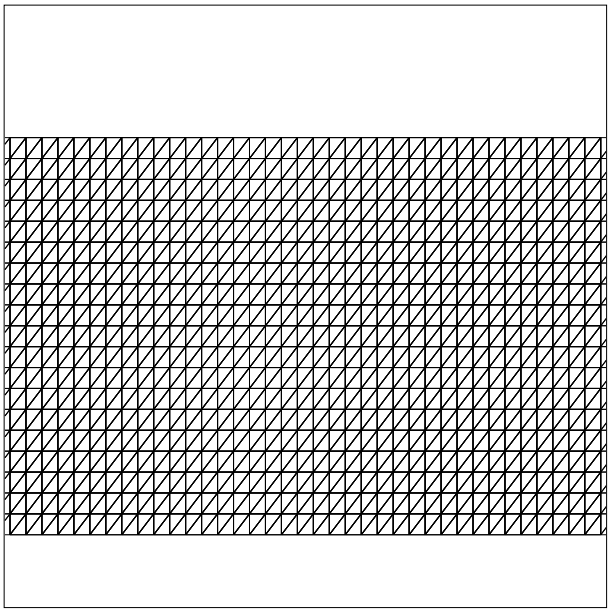

Figure 9: Structured triangular mesh of a rectangular domain

Table 1: Characteristics of the regular triangular meshes for the rectangular domain

Mesh	# Vertices	# Triangles	# Edges
RS1	4000	7562	11561
RS2	16000	31442	47441
RS3	64000	126962	190961

4.2.1 Flow inside a rectangular domain : regular triangulations

We consider the solution of the first linear system for several flow conditions corresponding to values of the freestream Mach number ranging from 0.1 to 0.9 and using a 2 subdomain decomposition of meshes RS1 to RS3 (see Tab. 1). The interface solver is the Richardson type algorithm 1. The results are summarized on Fig. 12 where we visualize, for each mesh, the required number of Richardson iterations to reduce the

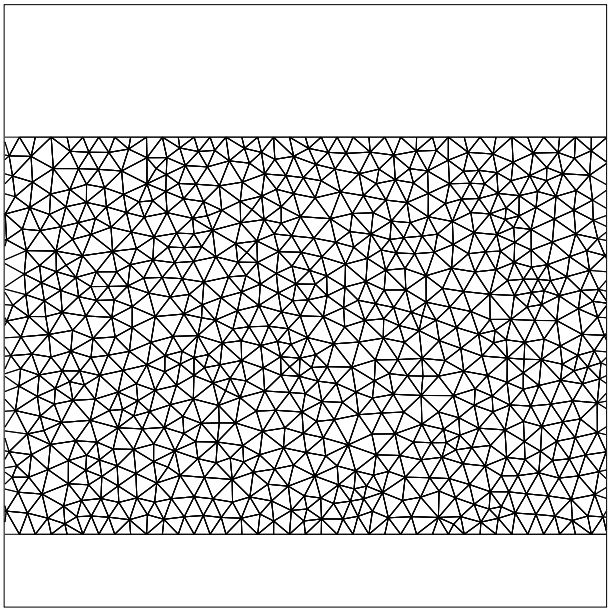

Figure 10: Unstructured triangular mesh of a rectangular domain

Table 2: Characteristics of the unstructures meshes for the rectangular domain

Mesh	# Vertices	# Triangles	# Edges
RU1	3740	7041	10780
RU2	14520	28164	42683
RU3	57203	112656	169858

Table 3: Characteristics of the meshes for the `NACA0012` airfoil

Mesh	# Vertices	# Triangles	# Edges
N1	3114	6056	9170
N2	12284	24224	36508
N3	48792	96896	145688

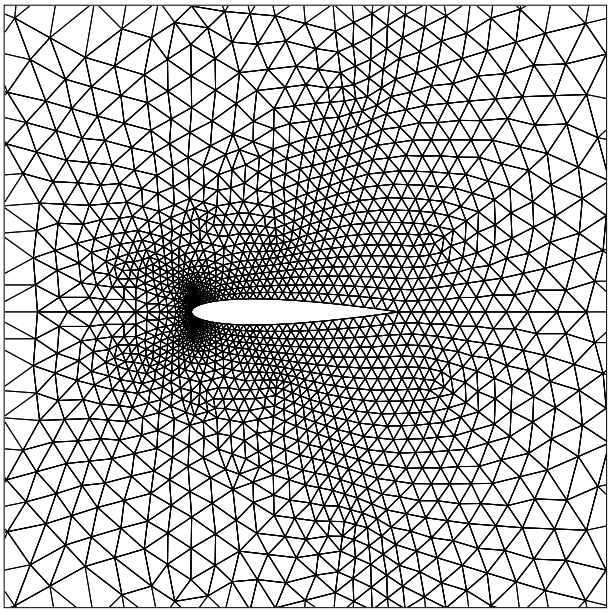

Figure 11: Unstructured triangular mesh around the NACA0012 airfoil

initial normalized residual to the threshold $\varepsilon_i = 10^{-10}$. From this first series of experiments we conclude that the observed convergence is at least qualitatively compatible with the results of section 3 (see Eq. 30). Moreover, the value $M = \sqrt{M_t^2 + M_n^2} = 0.6$ always yield the better convergence i.e. with the minimal number of iterations. Note that the value $u_0 = 1$ and the fact that we are using a regular triangulation of the rectangular domain, does not translate in $M_n = M$ and $M_t = 0$ because of the finite volume formulation adopted for the spatial approximation (see Fig. 8).

Fig. 13 visualize the convergence of the first linear system for $M = 0.3$ and for several decompositions in vertical stripes.

4.2.2 Flow inside a rectangular domain : unstructured triangulations

We consider the solution of the first linear system for several flow conditions corresponding to values of the freestream Mach number ranging from 0.1 to 0.9 and using a 2 subdomain decomposition of meshes RU1 to RU3 (see Tab. 2). The interface solver is the Richardson type algorithm 1. The results are summarized on Fig. 14 where we visualize, for each mesh, the required number of Richardson iterations to reduce the initial normalized residual to the threshold $\varepsilon_i = 10^{-10}$. This second series of experiments confirm the observation made previously. The value $M = \sqrt{M_t^2 + M_n^2} = 0.6$ is still

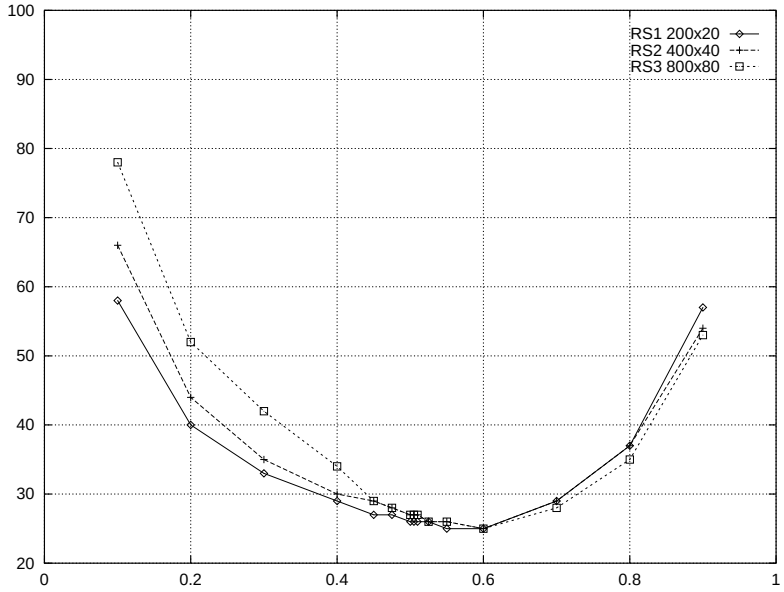

Figure 12: Convergence of the interface system
Flow inside a rectangular domain (regular triangulations) : 2 subdomain decomposition
X-axis : Mach number ($M = \sqrt{M_t^2 + M_n^2}$) - Y-axis : # Richardson iterations

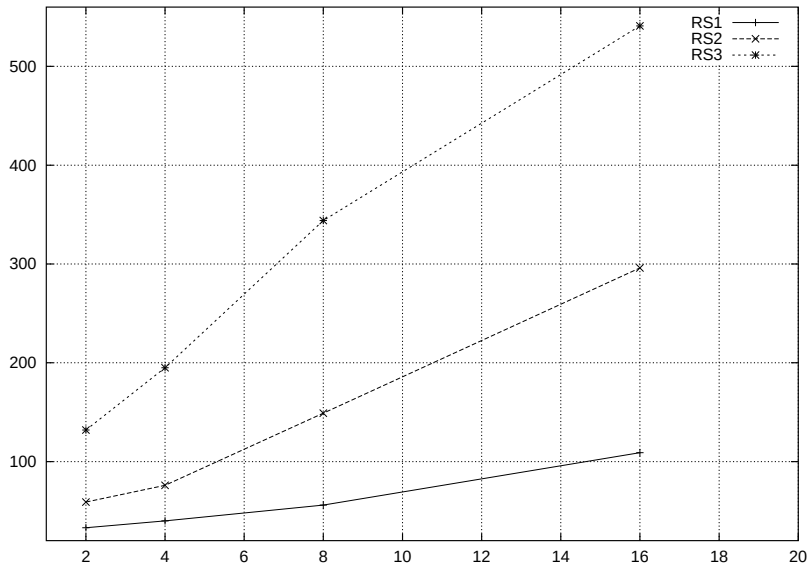

Figure 13: Convergence of the interface system : $M = \sqrt{M_t^2 + M_n^2} = 0.3$
Flow inside a rectangular domain (regular triangulations)
X-axis : # subdomains) - Y-axis : # Richardson iterations

the one that yield the better convergence i.e. with the minimal number of iterations. Fig. 15 visualize the convergence of the first linear system for $M = 0.3$ and for several decompositions in vertical stripes.

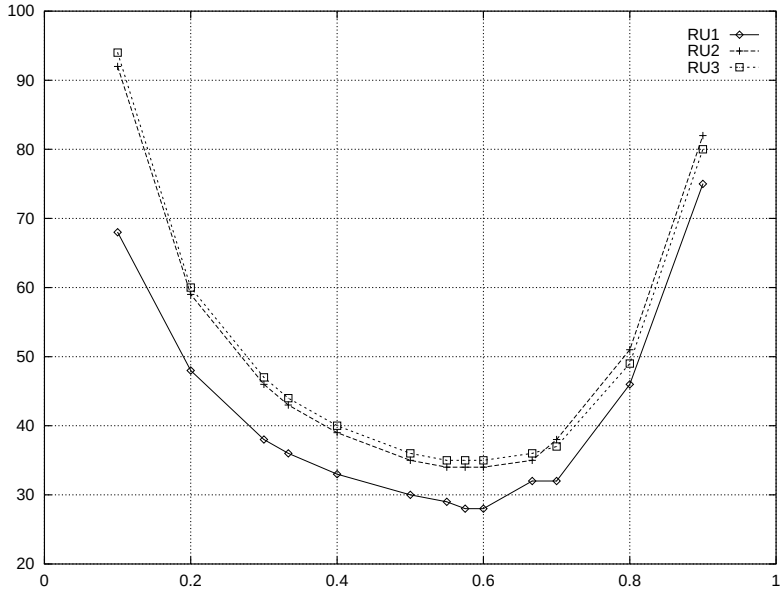

Figure 14: Convergence of the interface system
Flow inside a rectangular domain (unstructured triangulations) : 2 subdomain decomposition
X-axis : Mach number ($M = \sqrt{M_t^2 + M_n^2}$) - Y-axis : # Richardson iterations

4.2.3 Flow around a NACA0012 airfoil

We consider the solution of the first linear system for several flow conditions corresponding to values of the freestream Mach number ranging from 0.1 to 0.8 and using a 4 subdomain decomposition of meshes N1 to N3 (see Tab. 3). The interface solver is a full GMRES algorithm. The results are summarized on Fig. 16 where we visualize, for each mesh, the required number of GMRES iterations to reduce the initial normalized residual to the threshold $\varepsilon_i = 10^{-10}$. This time, a value of M slightly superior to 0.6 is observed to yield the better convergence. Finally, Fig. 17 compares the convergence of the first linear system for different decompositions of mesh N3 and using different solvers : a Richardson type algorithm on a 8 subdomain decomposition and a full GMRES algorithm on a 4 subdomain and a 8 subdomain decompositions.

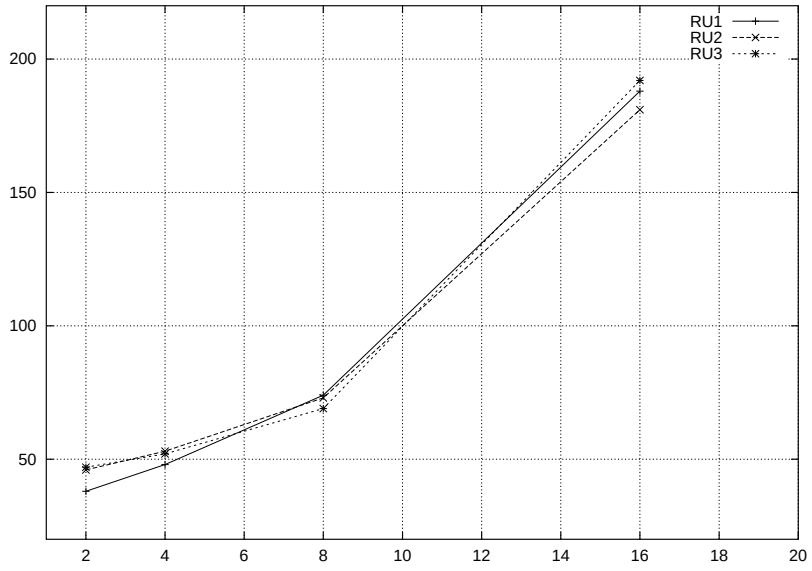

Figure 15: Convergence of the interface system : $M = \sqrt{M_t^2 + M_n^2} = 0.3$
Flow inside a rectangular domain (unstructured triangulations)
X-axis : # subdomains - Y-axis : # Richardson iterations

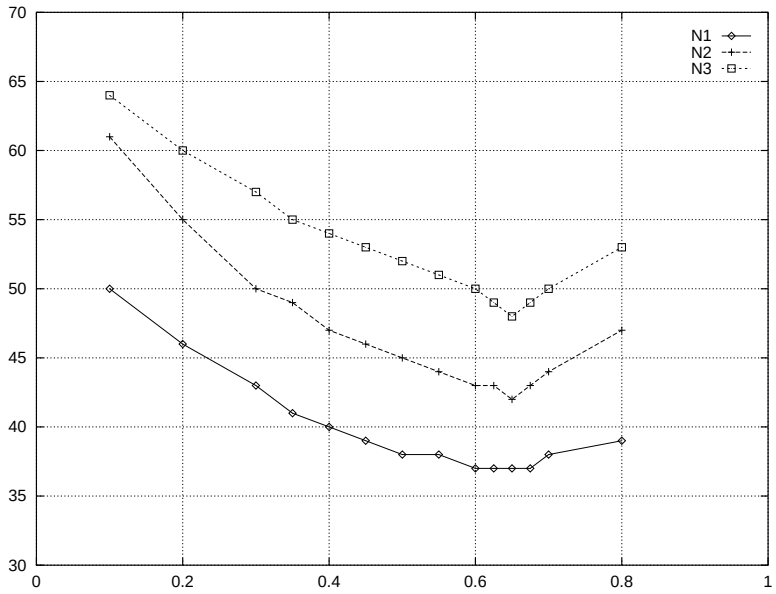

Figure 16: Convergence of the interface system
External flow around a `NACA0012` airfoil : 4 subdomain decomposition
X-axis : Mach number ($M = \sqrt{M_t^2 + M_n^2}$) - Y-axis : # GMRES iterations

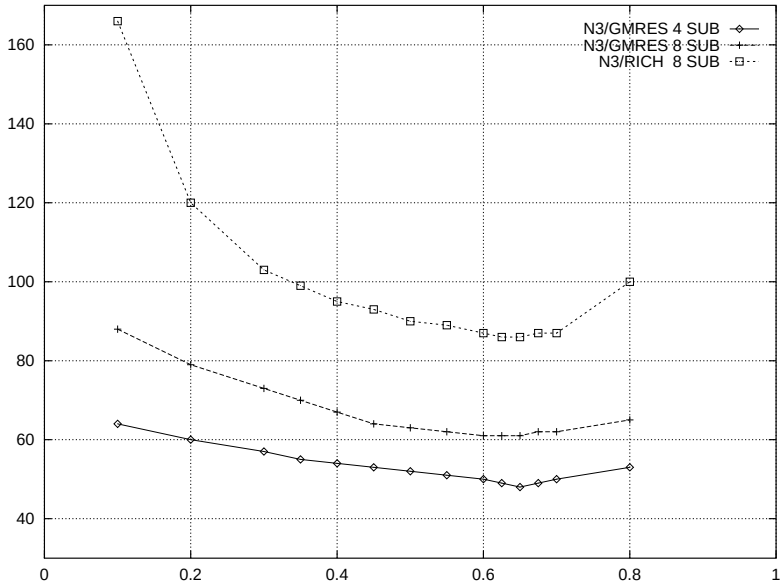

Figure 17: Convergence of the interface system
External flow around a `NACA0012` airfoil : mesh N3, 4 and 8 subdomain decomposition
X-axis : Mach number ($M = \sqrt{M_t^2 + M_n^2}$) - Y-axis : # GMRES/Richardson iterations

5 Conclusion

In this report, we have studied some aspects of a particular domain decomposition method applied to the solution of the Euler equations for compressible flows. The proposed method relies on the formulation of an additive Schwarz type algorithm on a non-overlapping decomposition of the computational domain. According to the hyperbolic nature of the Euler equations, the transmission conditions that are set at subdomain interfaces, express the conservation of the normal flux. Such conditions can be qualified as « classical interface conditions » in opposition to more sophisticated formulations such as the « optimized interface conditions » studied in [11] for the advection diffusion equation; for the two-dimensional Euler equations, the design of such optimized interface conditions is the object of an ongoing work[9]. The convergence of the method has been analyzed in the two- and three-dimensional cases, and for a two-subdomain decomposition, by considering the linearized equations and applying a Fourier analysis. Quite suprisingly, in spite of the fact that we use simple transmission conditions, the method converges and demonstrates an asymptotic convergence rate (i.e. for a large number of grid points) stricly inferior to 1. Various numerical experiments have confirmed at least qualitatively, the convergence behavior obtained analytically.

Following what has been presented here, our current investigations aim at assessing more deeply the mechanisms that make classical interface conditions sufficient to insure the convergence of the proposed non-overlapping additive Schwarz algorithm. In order to do so, we study the application of the so-called Smith factorization theory[7] instead of the usual diagonalization with eigenvalues/eigenvectors computation.

References

[1] T. Barth. *Numerical methods for gasdynamics systems on unstructured meshes*, volume 5 of *Lecture Notes in Computatational Science and Engineering*, pages 195–285. Springer Verlag, 1999.

[2] S. Clerc. *Etude de schémas décentrés implicites pour le calcul numérique en mécanique des fluides. Résolution par décomposition de domaine.* PhD thesis, Université de Paris VI, 1997.

[3] V. Dolean and S. Lanteri. A domain decomposition approach to finite volume solutions of the Euler equations on unstructured triangular meshes. Technical Report 3751, INRIA, 1999.

[4] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.*, 31:629–651, 1977.

[5] C. Farhat and S. Lanteri. Simulation of compressible viscous flows on a variety of mpps : computational algorithms for unstructured dynamic meshes and performance results. *Comp. Meth. in Appl. Mech. and Eng.*, 119:35–60, 1994.

[6] L. Fezoui and B. Stoufflet. A class of implicit upwind schemes for Euler simulations with unstructured meshes. *J. of Comp. Phys.*, 84:174–206, 1989.

[7] F.-R. Gantmacher. *Théorie des matrices.* Dunod, 1965.

[8] E. Godlewski and P.-A. Raviart. *Numerical approximation of hyperbolic systems of conservation laws*, volume 118 of *Applied Mathematical Sciences.* Springer Verlag, 1996.

[9] T. Gonzalez. *Parallélisation des schémas implicites de résolution des équations Euler compressibles par application des méthodes de décomposition de domaine.* PhD thesis, Université Paris VI, 2000.

[10] C. Japhet. *Méthode de décomposition de domaine et conditions aux limites artificielles en mécanique des fluides: Méthode Optimisée d'Ordre 2.* PhD thesis, Université Paris XIII, 1998.

[11] C. Japhet, F. Nataf, and F.X. Roux. The optimized order 2 method with a coarse grid preconditioner. application to convection-diffusion problems. In D. Keyes P. Bjorstad, M. Espedal, editor, *Proceedings of the 9th international conference on domain decompositon methods in science and engineering*, volume 218, pages 279–286. Wiley & Sons, 1998.

[12] D. Kroner. Absorbing boundary conditions for the linearized Euler equations in 2d. *Math. Comp.*, 57:153–167, 1991.

[13] M.-H. Lallemand, Steve S., and Dervieux A. Unstructured multigridding by volume agglomeration : current status. *Computers & Fluids*, 21:397–433, 1992.

[14] F. Nataf. On the use of open boundary conditions in block gauss-seidel methods for the convection-diffusion equation. Technical Report RI284, Centre de Mathématiques Appliquées, Ecole Polytechnique, 1993.

[15] F. Nataf and F. Nier. Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains. *Numer. Math.*, 75:357–377, 1997.

[16] F. Nataf, F. Rogier, and E. de Sturler. Optimal interface conditions for domain decomposition methods. Technical Report RI301, Centre de Mathématiques Appliquées, Ecole Polytechnique, 1994.

[17] A. Quarteroni and L. Stolcis. Homogeneous and heterogeneous domain decomposition methods for compressible flow at high reynolds numbers. Technical Report 33, CRS4, 1996.

[18] A. Quarteroni and A. Valli. *Domain decomposition methods for compressible flows*, pages 221–245. Kluwer Academic Publishers, 1999.

[19] P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

[20] Y. Saad and H. Schultz. Gmres : generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[21] B. Van Leer. Towards the ultimate conservative difference scheme V : a second-order sequel to Godunov's method. *J. of Comp. Phys.*, 32:361–370, 1979.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399